

普通高等教育“九五”国家教委重点教材

前　　言

在多年科学研究和教学实践中，我们深感应该有一本同时能适应数学系、计算机系和哲学系本科学生的逻辑教学用书。该书不仅向学生介绍数理逻辑的基本知识，而且在此基础上还能引导学生在某些方面走到研究的前沿，从而使他们具有更高、更深刻的观点，以利于他们的其他学科的学习。为此，四年前我们在南开大学数学专业试点班开设了这一课程，至今已讲过四次，受到同学们的欢迎。在该课进行中，我们仅见到一本类似的书，即："Computability：Computable Functions，Logic and Foundations of Mathematics（可计算性：可计算函数、逻辑和数学基础。详见引言）"。正如该书书名所表明的，该书和我们的书的内容有交叉，但侧重点有所不同。在此，我们希望有更多、更好的此类图书问世。另外，该书也可供相关专业的硕士生和教师参考，以补充他们的科研和教学的需要。

作者感谢曾在南开数学试点班修过该课的学生，他们中不少人对本书提出过修正意见。还特别感谢研究生丁龙云、李军、杨瑞光、李忠杰、李昭、唐鹏、武锦华、邵峰等，他们对本书提出过修正意见，有的还对本书的练习和参考文献进行过整理。本书作者徐书润也特别感谢他的老师唐稚松、陆钟万、张锦文、涂克仁等的早年教导；还感谢天津纺织学院潘家衍老师，他和作者进行过长期的、有关逻辑和集合论的十分有益的讨论。

由于水平和时间限制，本版书中错误和疏漏在所难免，恳请专家和读者指正。待有机会时再进行全面修订。

编著者

1997 年 8 月于南开

目　　录

第一章 引　　言

首先，介绍一下本书的题目。

这里所谓的“计算机”，并不是单指通常作各种科学计算或非数值计算的电子计算机，而是包括各种各样的计算装置和在其上实现的计算。用一个更恰当的词，称作“理想计算机”，这即通常所说的“可计算理论”。

这里所谓的“逻辑”是专指“数理逻辑”，即用数学方法来研究的数学中的逻辑。因而，既不是指生活中的“逻辑”，也不是指目前哲学中令人难以琢磨的“逻辑”。

这里所谓的“集合论”是广义的，只不过在本文中是从数学角度上对其进行抽象的研究。当然，我们既要讨论所谓的朴素集合论，还要讨论公理化集合论，以及 Zermelo-Fraenkel 公理集合论系统 ZF。

其次，我们采取追根溯源的办法，介绍计算机、逻辑和集合论这三者之间的关系。以下分 3 个题目来讲：

1. 集合论和第三次数学危机

集合论（或更确切地说是朴素集合论）是在 19 世纪 70 年代建立起来的。具体说来，1873 年 12 月，G. Cantor 发现实数是不可数的，并写信告诉了 Dedekind；次年，Cantor 发表了第一篇有关的研究论文。1882 年 Cantor 宣布他证明了连续统猜想（简记为 CH），但直至他 1918 年去世也一直未能发表，且在他去世之前曾很遗憾地表示，自己未能最终完成这一证明。尽管如此，在他的开创性工作导引下，集合论已经是成果卓著，并可以作为各门数学的基础了。所以在 1900 年国际数学家大会上，Poincalé 宣称：数学大厦已经建成。然而，就在两年之后，即出现了 Russell 的有名

的悖论（X 为所有集合之集合）。这一悖论，在 Cantor 给 Dedekind 的信中也已考虑到，但无好的解决方法。再加上其他一些类似的悖论先后出现，于是便导致了第三次数学危机。第一次数学危机是无理数的出现，第二次则是有关如何理解高阶无穷小。可见，这三次危机均是与数学中的无穷相关的。为解决这第三次危机，本世纪前 30 年出现了有名的三大主义，即以 Russell 为代表的逻辑主义、以 Hilbert 为代表的形式主义和以 Brouwer 为代表的直觉主义。他们分别从各自的观点出发，试图克服这次危机。特别是逻辑主义和形式主义，最后都是从逻辑出发，建立多种类逻辑或用逻辑对集合论给以公理化处理。这便引起我们第二部分要讨论的内容。

2. 逻辑与有穷性（finitary）原则

逻辑（即英文 mathematical logic）有两个来源：一是日常生活中的逻辑，一是数学中的逻辑。所谓数理逻辑即是“用数学的方法来研究数学中的逻辑”。早在 17 世纪的 Leibniz，即试图设计一种语言，用它来书写数学证明，且按这种形式证明，不用加任何其他解释，即可机械地判定此证明有无错误，而不会出现任何争议。但是，具有先见性的 Leibniz 仅仅有这样一个想法，而真能称得起是命题逻辑的，则要到 19 世纪中期的 Boole，而更能称得起是数学逻辑的，则要到 1879 年 Frege 的谓词逻辑。此后，用谓词逻辑描述初等几何和算术（＋，·），如 Hilbert 的《几何基础》中所阐述的内容等。

当第三次数学危机出现后，逻辑便在严格的数学基础中发挥了作用。前面所说的逻辑主义和形式主义，可以说是殊途同归。与此相应的是对朴素集合论进行公理化。其中主要的有两种：一是 1904 年 Zermelo 关于公理集论的研究，以及后来与 Frankel 一起用谓词逻辑表述的 ZF 系统（或者再加上选择公理 AC 之后称之为的 ZFC）；二是所谓的 GBN 系统，由 Gödel，Bernays 和 von Neumann 等人建立和发展。根据这些公理化集论的研究，确实不

会产生什么矛盾，但并没有真正地刻画了“集合”涵义。这个结论可从 Gödel 的 1931 年的形式数论系统不完全得到。而到 1963 年，由 P. J. Cohen 证得 CH 是独立于 ZF 的，其证明方法即所谓的力迫法，正是引入可以是也可以不是集合的对象作为集合而加以证明的。

那么，这到底是什么原因？还是 Gödel 及其之后的研究者（如 M. Davis 等）所指出的，这是由于坚持了所谓的“有穷性原则”，即一方面是将表示数学内容的符号和其意义分开，另一方面是数学公式的有穷长及推理规则的有穷可判定。为阐述所谓的有穷性原则，我们讨论下面的第三部分内容。

3. 有穷性原则和理想计算机

自 1900 年 Hilbert 提出 23 个数学问题向 20 世纪数学家挑战之后，不少人对其中的第十问题进行研究。该问题是说：有无一个确定的方法，使对任给的 Diophanto 方程（即整系数多项式方程），可在有穷步判定其有无正整数解。起初，大部分研究者认为是可以找到这样的方法的。但经 30 多年努力，均告失败后，有些人就觉得：可能根本不存在这样的方法。但要证明这一想法是正确的，则必须首先说明究竟什么是“方法”。这便是 20 世纪 30 年代中期开始的各种可计算理论的研究。其中，最著名的是：Gödel、Kleene 等的递归函数论，A. Turing 的 Turing 机器理论，A. Church 的 λ-演算理论，Post 系统，以及 50 年代的 Markov 算法论、递归算法论（胡世华）、URIM（Minsky 等）。这些，都是对“方法”所作的不同定义，但能证明这种种定义在一定意义上又是相互等价的。由此，再加上一直未找到任何一个不包含在上述任何一种定义中的“方法”，故就产生出一种信念：任何“方法”不过是由上述所定义的。这便是所谓的 Church-Turing 论题（任何一个可计算函数均可由 Turing 机计算）。在这基础上，经 M. Davis，J. Robinson 和 H. Putnam 等人多年的工作，在 1970 年，由前苏联年轻的数学家 Matijasevič（当时还是学生），证得了 Hilbert 第

十问题是不可解的，即不存在一个确定的方法（或严格地说是不存在一个 Turing 机），使对任给的 Diophanto 方程，可在有穷步判定其有无整数解。

至 70 年代，随着计算机及其应用的广泛发展，有些科学家开始研究理想计算机的由来，并得出结论：上述种种定义的实质是所谓的“有穷性”和“确定性”（如王浩等）。胡国定教授和我就是在上述研究的基础上，结合了 Turing 机的确定性（排除了其对运算对象操作的特殊性）和 Post 系统对运算对象操作的一般性（排除了它的非确定性），对上述“方法”，作了最一般的定义，即所谓的计算机的数学模型（MMCM），而且进一步明确指出：由这种“方法”所确定的原则即所谓的“有穷性原则”。

由上述集合论、逻辑和计算机的关系，可确定进行下述从计算机（有穷性原则）到逻辑，进而再到集合论的研究。

那么，到底有什么重大问题值得特别提出和讨论呢？这就是从

$$\text{朴素集合论} \Longrightarrow \text{ZF 系统(或 ZFC 系统)}$$

产生的很多问题。比如 CH 应要么为真，要么为假，可是 ZF 却不能确定其真、假性（或称 CH 独立于 ZF）。造成这种情况的原因是集合论公理不够，比如大基数问题、决定性公理等的研究都是与此相关的。正是由于这些问题，刺激了集合论的发展，在这不断深入的研究过程中，人们发现了不少新方法和新观点，从而必将把我们带入更广阔更自由的境界。

造成这一情况的另一原因是一阶逻辑的有穷性原则。为了克服所存在的缺点，从 50 年代至今，发展了很多逻辑。这里不是指在计算机科学中应用的种种更局限、更有新特色的逻辑，而是指更广泛的、更适于描述数学的逻辑。J. Barwise 在他 1985 年所编的书“Model Theoretic Logics”中，介绍了 30 多位数理逻辑学家的有关研究结果。S. Shapiro 在其 1991 年的书“Foundations Without Foundationalism”中，将二阶逻辑作了更详细的论述。本书将摘要介绍其中一些重要结果。此处特别提出，近几年，胡国

定建立的一般逻辑是有相当特色的。该种逻辑，针对有穷性原则，不再将符号和其涵义分开，并承认朴素集合论的基本内容（为免除悖论，稍加修改），以此为出发点建立数学基础。

理想计算机领域的问题就更多了。几乎可以说是俯拾即是。计算机的预研、设计、研制和应用，包括硬件和软件，向科研人员提出一系列新的理论问题。正是这些问题的研究和解决，推动着计算机科学飞速发展。比如，已经30多年了，尚未解决的“P=NP?”问题，引起全世界理论计算机科学家和数学家的关心，有些人甚至潜心竭虑，可至今尚看不出解决的眉目。有多种著名期刊设专栏，刊载有关这方面研究的进展。早在10年前，就有人将该问题与CH问题作对比。两个很不相同的领域的问题，表面看来十分不同，然而也有很多相似之处。这不能不说，科学是多么的奇妙诱人。有心计的科学家和研究者，一定会在这些问题的解决中，贡献出他们的聪明和才智。

本书是写给大学数学系和计算机系的学生用的。对数学系的学生，本书将起到向实际方面引导的作用，且又能使他们受到在其他数学中未介绍过的形式方法的训练，特别是集合论使得他们对所研究的数学有了更牢靠的基础。对计算机系的学生，本书能引导他们使用一般的方法，开启他们解决实际问题的思路，并能从宏观上对自己所学和所作的，有更深刻、更概括的观点，从而有助于新概念、新方法的建立。本书也可作为有关领域的教师和研究者的参考用书。因为本书不仅有基本的内容介绍，还有最新的研究报道，这些内容既可作为他们教学和研究的补充，也可将他们带入有兴趣的研究前沿。

在1989年，美国著名的逻辑学家、数学家Epstein等出版一书，名为“Computability: Computable Functions, Logic and the Foundations of Mathematics”（可计算性：可计算函数、逻辑和数学基础）。该书内容和本书有所交叉。这也如其书名一样，与本书相同的是前二者，而第三者有所不同。他们的这本书是一种系列丛书的一本。丛书由哈佛、MIT、密执根、芝加哥等大学的名教

授主编。

最后，简要介绍一下本书的内容。正如前面所说，本书包括理想计算机、逻辑和集合论三部分内容。实际上，这已包含了数理逻辑的四大论（即证明论、递归论、集合论和模型论）的基本内容。当然，如王浩所说，数理逻辑的两个不确定的领域，即数学基础和计算机科学，该书中也有所涉及。具体言之，本书第二章为理想计算机与有穷性原则。该章一开始就指出，计算机的能力是和计算装置、计算机程序设计语言、计算机程序及其运行直接相关的。为了给出刻画计算能力的标准，该章接着介绍了递归函数及其子类，这也为讨论计算复杂性打下基础。然后，该章通过种种有关计算的实例，分析综合出其中的要素，并结合已定义的理想计算机，如 Turing 机和 Post 系统，定义出最一般的理想计算机 MMCM（计算机的数学模型）。接着用 MMCM 研究了递归集、递归可枚举集。并最后指出：有穷性原则就是由有穷映射定义递归集和递归可枚举集的原则。本书第三章是有穷性逻辑和有穷性数学。该章先简述数理逻辑产生和发展的历史，接着以群为例子，介绍谓词逻辑的语法、语义和推理规则。然后，全面而且扼要地介绍了一阶逻辑，并证明完全性定理等有关的重要结果。该章还讲解了 Gödel 不完全性定理的实质，分析了一阶逻辑的局限性，并简要介绍了二阶逻辑和 Q 量词逻辑。再后，作为一阶逻辑、二阶逻辑和 Q 量词逻辑的抽象，该章介绍了有穷性逻辑和有穷性数学，并研究了它的简单性质。为了克服有穷性逻辑的缺点，该章还介绍了无穷逻辑 $L_{\omega_1\omega}$，并提出如何建立既包括二阶逻辑又包括 $L_{\omega_1\omega}$的问题。为了下一章更顺利地介绍一般逻辑和一般数学，该章最后着重介绍无穷命题演算，且为了作对照，还介绍了有穷命题演算，特别是王浩的命题逻辑系统P_{ω}。本书第四章是一般逻辑和一般数学。这章介绍的是胡国定近几年的研究结果。这里，一般逻辑和一般数学与第三章所讨论的种种逻辑的最大不同有二点：即不再将语法和语义分开，并且可随意使用 Cantor 集合论。该章首先介绍了一般逻辑和一般数学的定义，并以多个例子说明该

逻辑中是如何进行证明的。然后介绍了如何对一般逻辑和一般数学作语义解释，并证明了语义完全性定理。该章最后一节，作者对第三章和第四章的种种逻辑作了总回顾，并作了一定的对比，提出一些观点。第五章是集合论，这是该书的最后一章。该章首先以序数、基数为基线，比较详尽地介绍了朴素集合论。该部分的介绍使得不熟悉集合论的读者也能读懂。该章然后介绍了公理集合论。这种集合论的公理，是用自然语言叙述的。不少讲公理集合论的书上，开始也是这样讲解的，但仍然以一阶公理集合论，或称作 ZF 系统为主。而这里，确确实实与 ZF 是不同的，并作了明显的对照。鉴于朴素集合论已讲得相当详尽了，故为了避免重复，只是对序数和基数作了些介绍，而且这里的序数和基数是等价类的典型代表，不像朴素集合论中是等价类。该章最后介绍了 ZF 系统，主要是讲述选择公理的几种等价形式，并重点讲了 Gödel 关于 ZF 与连续统猜想、选择公理相协调的证明。鉴于本书的性质，并未给出十分严格的证明。该章最后介绍了 Cohen 关于 ZF 与连续统猜想、选择公理二者的否定命题相协调的证明思想，从而最后得到有名的独立性结果：CH，AC 独立于 ZF。

第二章　理想计算机与有穷性原则

计算机已相当普及。计算机能作各种各样的计算，有数值的、有非数值的；特别是计算机还可用来作控制、辅助设计，甚至定理证明和逻辑推理。那么，到底计算机的能力有多大？

1945 年世界上第一台电子计算机出现以来，已制造出了千千万万台计算机，有电子管的、晶体管的或集成电路的；目前还正在研究光子计算机和分子生物计算机。特别是已有用十多万个处理器组装起来的巨型计算机。那么，到底人类还能造出什么样的功能更强的计算机？

众所周知，要使用这种种计算机，必须要有各种与机器相配的程序设计语言，并使用该语言编制各种功能的程序。当运行这些程序时，计算机就执行这些程序所规定的动作，一步步地实现该程序的功能。

从上述简单的讨论不难看到，计算机的能力是和计算机装置、计算机程序设计语言、计算机程序直接相关的。

本章就是要通过种种有关计算的实例，分析综合出其中的要素，并结合已定义的理想计算机，如 Turing 机和 Post 系统，定义出最一般的理想计算机 MMCM（计算机的数学模型）。为了给出刻画计算能力的标准，本章先介绍递归函数及其子类，这也为进一步研究计算复杂性打下基础。本章最后指出**有穷性原则**就是由**有穷映射定义递归集和递归可枚举集**的原则。

第一节　递 归 函 数

递归函数是定义在自然数集 $N=\{0, 1, 2, 3, \cdots\}$ 上的函数。这种函数是通过归纳而定义的。即先假定若干个已知的函数

（它们的定义不必再进一步追究，事实上它们都是十分明显的），而后再给出几个由已定义的函数构造新函数的算子；并且称这若干个**已知函数为初始函数，构造新函数的算子则称作函数构造算子。从初始函数出发，由函数构造算子不断构造出的所有函数即是这里所谓的递归函数（新构造出的函数仍然可使用函数构造算子不断构造出更新的函数）**。

1.1 原始递归函数集

原始递归函数集是递归函数集的真子集，如下定义之。

初始函数：

$S(x) = x+1$，　　称作后继函数；

$Z(x) = 0$，　　称作零函数；

$U_i^n(x_1,\cdots,x_n) = x_i$，　　称作射影函数（对任 $n \in N, 1 \leqslant i \leqslant n$）。

函数构造算子：包括代入和递归两个算子。**代入算子**：设 $g_1(x_1,\cdots,x_n), g_2(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n), h(y_1,\cdots,y_m)$为已定义好的函数，则

$$f(x_1,\cdots,x_n) = h(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$$

为由代入算子新定义的函数；

递归算子：设 $g(x_1,\cdots,x_n), h(y_1,\cdots,y_{n+2})$为已定义好的函数，则如下定义的 $f(x_1,\cdots x_n,y)$为由递归算子新定义的函数

$$\begin{cases} f(x_1,\cdots,x_n,0) = g(x_1,\cdots,x_n) \\ f(x_1,\cdots,x_n,y+1) = h(x_1,\cdots,x_n,y,f(x_1,\cdots,x_n,y))。\end{cases}$$

那么，**原始递归函数集就是从上述初始函数出发，并经上述二个算子一步步构造出的所有函数所构成之集合**。

这里，我们解释一下递归算子。使用该算子所定义的新函数 $f(x_1,\cdots,x_n,y)$是归纳于其变元 y 进行定义的。就是说，

$$\begin{aligned} f(x_1,\cdots,x_n,0) &= g(x_1,\cdots,x_n), \\ f(x_1,\cdots,x_n,1) &= h(x_1,\cdots,x_n,0,f(x_1,\cdots,x_n,0)) \\ &= h(x_1,\cdots,x_n,0,g(x_1,\cdots,x_n)) \end{aligned}$$

$$f(x_1,\cdots,x_n,2)=h(x_1,\cdots,x_n,1,f(x_1,\cdots,x_n,1))$$
$$=h(x_1,\cdots,x_n,1,h(x_1,\cdots,x_n,0,g(x_1,\cdots,x_n)))$$
$$\vdots$$

由此,并注意到代入算子,则计算任给的原始递归函数时,均可最终化归到计算初始函数的值,也就是说,**当假设初始函数对任给的自变量之值,均能求得其函数值时,则任给的原始递归函数,对任给的自变量之值,也均能一步步地求得该函数的值。**

现在研究一些原始递归函数。

例 1 $x+y$ 是原始递归函数。因为

$$\begin{cases}x+0=U_1^1(x)\\ x+(y+1)=h(x,y,x+y),\end{cases}$$

$$\text{且 } h(x,y,z)=S(U_3^3(x,y,z))。$$

例 2 $x\cdot y$ 是原始递归函数。因为

$$\begin{cases}x\cdot 0=Z(x)\\ x\cdot(y+1)=h(x,y,x\cdot y),\end{cases}$$

$$\text{且 } h(x,y,z)=U_3^3(x,y,z)+U_1^3(x,y,z)。$$

例 3 x^y 是原始递归函数。留作练习。

例 4 先驱函数 $pd(x)$(当 $x=0$ 时,$pd(x)=0$,当 $x\geqslant 1$ 时,$pd(x)=x-1$)也是原始递归函数。因为

$$pd(x)=pd_1(x,x)=pd_1(U_1^1(x),U_1^1(x)),$$

且 $pd_1(x,y)$如下定义:

$$\begin{cases}pd_1(x,0)=Z(x)\\ pd_1(x,y+1)=U_2^3(x,y,pd_1(x,y))。\end{cases}$$

例 5—例 11 下述函数均为原始递归函数。留作练习。

$x\dot{-}y$ (非负减.当 $x\leqslant y$ 时,$x\dot{-}y=0$;当 $x>y$ 时,$x\dot{-}y=x-y$)

$|x-y|$(绝对值减)

$S_g(x)$(符号函数。当 $x=0$ 时,$S_g(x)=0$;当 $x>0$ 时,$S_g(x)=$

1)

$\overline{S_g}(x)$（反符号函数。当 $x=0$ 时，$\overline{S_g}(x)=1$；而当 $x>0$ 时，$\overline{S_g}(x)=0$）

$rem(x,y)$（x 除以 y 的余数，且约定当 $y=0$ 时，$rem(x,y)=x$）

$[x/y]$（x 除以 y 的整部，且约定当 $y=0$ 时，$[x/y]=0$）

$div(x,y)$（当 $x,y>0$ 且 $x\ \ y$ 时，则 $dir(x,y)=1$；否则，$div(x,y)=0$。）

1.2 原始递归算子

凡由**已知的原始递归函数构造出新的原始递归函数的算子，我们称之为原始递归算子**。当然，上述定义原始递归函数所用的代入和递归算子是原始递归算子。连加和连乘也是原始递归算子。

定理 1 设连加、连乘算子为

$$g(x_1,\cdots,x_n,y)=\sum_{z=0}^{y}f(x_1,\cdots,x_n,z),$$

$$g(x_1,\cdots,x_n,y)=\prod_{z=0}^{y}f(x_1,\cdots,x_n,z),$$

则它们均为原始递归算子。

证明（略）留作练习。

现在，再介绍二个原始递归算子，即**受囿递归算子**和**受囿最小求根算子**（后者也称作**受囿 μ-算子**）。

设 $g(x_1,\cdots,x_n),h(x_1,\cdots,x_{n+2}),i(x_1,\cdots,x_n,y)$ 是原始递归函数，令

$$\begin{cases}f(x_1,\cdots,x_n,0)=g(x_1,\cdots,x_n)\\ f(x_1,\cdots,x_n,y+1)=h(x_1,\cdots,x_n,y,f(x_1,\cdots,x_n,y)\\ f(x_1,\cdots,x_n,y)\leqslant i(x_1,\cdots,x_n,y),\end{cases}$$

则称 **f 为由 g,h,i 经受囿递归算子构造出来**。易见，这与递归算子的区别仅仅是多了第三条定义式，即当一个新的函数 f 被通过递归算子构造出时，必须保证有一个已知的函数 i（即已定义好的函

数)，使得 f 受囿于该 i（即总有 $f \leqslant i$）。

设 $g(x_1, \cdots, x_n, y)$ 是原始递归函数，令

$$f(x_1, \cdots, x_n, z) = \mu y_{\leqslant z}[g(x_1, \cdots, x_n, y) = 0],$$

即当有 $y \leqslant z$，使得 $g(x_1, \cdots, x_n, y) = 0$ 时，则 $f(x_1, \cdots, x_n, z)$ 为最小的这种 y；否则，$f(x_1, \cdots, x_n, z) = z$。读作"$f$ 等于最小的 $y \leqslant z$，使 $g(x_1, \cdots, x_n, y) = 0$"。这即所谓的**受囿最小求根算子**。

那么，我们有：

定理 2 受囿递归算子和受囿最小求根算子是原始递归算子。

证明 前者显然。只证后者。事实上，

$$f(x_1, \cdots, x_n, z) = \left(\sum_{k=0}^{pd(z)} \prod_{y=0}^{k} \mathrm{S_g}(g(x_1, \cdots, x_n, y))\right) \cdot \mathrm{S_g}(z),$$

而由例 4，7 和定理 1，结合原始递归函数定义，立得所证。

最后，讨论**分情形定义算子**。

设 $g_i(x_1, \cdots, x_n)(i = 1, \cdots, m+1)$ 和 $C_j(x_1, \cdots, x_n)(j = 1, \cdots, m)$ 均为原始递归函数，令

$$f(x_1, \cdots, x_n) = \begin{cases} g_1(x_1, \cdots, x_n), \text{当 } C_1(x_1, \cdots, x_n) = 0 \text{ 时}, \\ g_2(x_1, \cdots, x_n), \text{当 } C_2(x_1, \cdots, x_n) = 0 \text{ 时}, \\ \quad \vdots \\ g_m(x_1, \cdots, x_n), \text{当 } C_m(x_1, \cdots, x_n) = 0 \text{ 时}, \\ g_{m+1}(x_1, \cdots, x_n), \text{否则}, \end{cases}$$

（且这里的 C_j 是两两互斥的，即对给定的 $x_1, \cdots, x_n$，最多只有一个 $C_j(x_1, \cdots, x_n) = 0$），则称**函数 f 是由函数 g_i 和 C_j 经分情形算子构造的**。我们有：

定理 3 分情形定义算子是原始递归算子。

证明 留作练习。

例 12 设 $F(x) = \sum_{k=0}^{x} \mathrm{div}(k, x)$，$F(x)$ 自然是原始递归函数；其值为 x 的因子个数。

例 13 设 $Pr(x) = \overline{\mathrm{Sg}}(|F(x) - 2|)$，$Pr(x)$ 自然也为原始递

归函数；其值为 1 或 0；且当 x 为素数时，$Pr(x)$为 1；否则为 0。

例 14 设 $P_m(x)=\sum_{k=0}^{x}S_g(Pr(k))$，$P_m(x)$自然为原始递归函数；其值为$\leqslant x$ 的素数的个数。

例 15 设 P_k 表示第 $k+1$ 个素数，即

$$P_0=2,P_1=3,P_2=5,P_3=7,\cdots,$$

那么，注意到初等数论中的结果：在 P_k 和 $2P_k$ 间肯定有 P_{k+1}（见《数论导引》pp. 97—99），故可如下定义 P_k，

$$\begin{cases}P_0=2\\ P_{k+1}=\mu y_{\leqslant 2P_k}[(k+2)\dot{-}P_m(y)=0],\end{cases}$$

试严格证明 P_k 是原始递归函数（留作练习）。上述第 2 式也可直接定义为：

$$P_{k+1}=\mu y_{\leqslant 2P_k}[(1\dot{-}P_r(y))+\overline{S_g}(y\dot{-}P_k)=0]。$$

例 16 设 $\ln(0)=\ln(1)=0$，

$\ln(x+2)=\mu z_{\leqslant x+2}[z\neq 0$，且 $\operatorname{div}(P_{z\dot{-}1},x+2)=1$，且对任 $z\leqslant i$
$<x+2,\operatorname{div}(P_i,x+2)=0]$，

试说明 $\ln(x)$的涵义，并严格证明其原始递归。

例 17 设$(x)_i=\mu y_{\leqslant x}[\overline{S_g}(\operatorname{div}(p_i^y,x))+\operatorname{div}(P_i^{y+1},x)=0]$，其中 i 为参量。易知，$(x)_i$ 是原始递归函数；且当 $x\geqslant 2$ 时，该函数的值为 x 所含因子 P_i 的最多个数。

1.3 原始递归函数集的分层

Grzegorczyk 在其 1953 年的文章中（见《数理逻辑论文选》pp. 232—274），对原始递归函数集作了一个完全的层次性的分类，即：设 $\mathfrak{R}$ 为原始递归函数集，则有 $\mathfrak{R}$ 的子集 $\varepsilon^i(i=0,1,2,\cdots)$，使得(1)$\varepsilon^i\subsetneqq\varepsilon^{i+1}(i=0,1,\cdots)$，(2)$\mathfrak{R}=\bigcup_{i=0}^{\infty}\varepsilon^i$. 这样，**当我们研究各种理想计算机的功能时，即可将它们所能计算的函数集合与各 ε 进行比较，从而即可从某种角度上确定这些计算机的能力强或弱。**

这里，首先定义 **Grzegorczyk** 函数序列 $f_i(x,y)$：

$$f_0(x,y) = y + 1$$
$$f_1(x,y) = x + y$$
$$f_2(x,y) = (x+1)(y+1)$$
对任 $i \geqslant 2$，
$$\begin{cases} f_{i+1}(0,y) = f_i(y+1,y+1) \\ f_{i+1}(x+1,y) = f_{i+1}(x,f_{i+1}(x,y)). \end{cases}$$

这里定义的 $f_i(i \geqslant 3)$，不是使用的前述简单的递归算子，而是**单重嵌套递归算子的一种特殊情形。单重嵌套递归算子的最一般情形为：**
$$\begin{cases} f(0,x) = g(x) \\ f(y',x) = h(y,f(y,i(y,x,f(y,x))))。\end{cases}$$
但是，可以证明：单重嵌套递归算子也是一种原始递归算子。因而，任 f_i 均为原始递归函数。适当时候将介绍全部证明。这里仅看看 f_3 的计算过程：
$$f_3(0,y) = f_2(y+1,y+1) = (y+2)^2$$
$$f_3(1,y) = f_3(0,f_3(0,y)) = (f_3(0,y)+2)^2$$
$$= ((y+2)^2+2)^2$$
$$f_3(2,y) = f_3(1,f_3(1,y))$$
$$= ((f_3(1,y)+2)^2+2)^2$$
$$= ((((y+2)^2+2)^2+2)^2+2)^2$$
$$f_3(3,y) = \underbrace{(\cdots}_{8\text{个}}(y+2)^2+\cdots+2)^2$$
$$\vdots$$

现在来定义 **Grzegorczyk 函数类** $\varepsilon^i(i \geqslant 0)$：

初始函数：$S(x)$，$U_1^2(x,y)$，$U_2^2(x,y)$，$f_i(x,y)$，$Z(x)$；

函数构造算子：代入算子，受囿递归算子；那么，ε^i 就是从上述初始函出发，并经由上述函数构造算子一步步得到的所有函数所成之集。

定理 4 设 $\Re$ 是原始递归函数集，$\varepsilon^i(i \geqslant 0)$是 Grzegorczyk 函

数类，则有

(1)$\varepsilon^i \subsetneq \varepsilon^{i+1}, i \geqslant 0$；　(2)$\mathscr{R} = \bigcup_{i=0}^{\infty} \varepsilon^i$。

证明　事实上，$f_{i+1} \in \varepsilon^i$，$\therefore x+y \in \varepsilon^0, x \cdot y \in \varepsilon^1$。详略。

1.4　Arckermann 函数

从 **1.1** 已看出，任何原始递归函数都是可一步步求得其值的，只要给定了自变量值，且又假设初始函数是能求得它们的值的话。那么，是否还有可一步步求得其值的函数，而它又非原始递归的呢？回答是肯定的。非常著名且有用处的 Arckermann 函数就是一例。其定义为：

$$\begin{cases} A(0,y) = y+1, \\ A(x+1,0) = A(x,1), \\ A(x+1,y+1) = A(x,A(x+1,y))。 \end{cases}$$

这里定义 Arckermann 函数所用的函数构造算子，既非简单的递归算子，也非 **1.3** 中所介绍的单重嵌套递归算子，而是所谓的**双重嵌套递归算子**，即是说定义 $A(x,y)$是归纳于它的二个变元，而不是如 Grzegorczyk 函数那样仅仅归纳于一个变元。顺便指出，Grzegorczyk 函数倒是参照 Arckermann 函数定义出的。

定理 5　$A(x,y)$满足：

(1) $y < A(x,y)$；

(2) $A(x,y) < A(x,y+1)$；

(3) $A(x,y+1) \leqslant A(x+1,y)$；

(4) $A(x,y) < A(x+1,y)$；

(5) 对任给定的 $c_1,\cdots,c_r$，均有 $c = \max(c_1,\cdots,c_r) + 4r$，

使 $\sum_{i=1}^{r} A(c_i,y) \leqslant A(c,y)$。

证明　略。有兴趣者试证明之。

定理 6　对任给原始递归函数 $f(x_1,\cdots,x_n)$，均可找到常数 c，使

$$f(x_1,\cdots,x_n) < A(c,x_1+\cdots+x_n)。$$

证明　要证明该定理，必须施归纳于原始递归函数的构造，即先对已知的初始函数证明定理满足，而后对各函数构造算子逐一进行归纳证明之，即假设已知的函数满足该定理，然后证明由函数构造算子定义的新函数也满足该定理。详细证明略，读者试补证之。

定理 7　$A(x,y)$非原始递归函数。

证明　若$A(x,y)$是原始递归函数，则$A(x,x)$显然也为原始递归函数。那么，由定理 6，必可找到常数c，使$A(x,x)<A(c,x)$对任何x均成立。但若取$x=c$，即得$A(c,c)<A(c,c)$，矛盾。故证明所设为错，即$A(x,y)$非原始递归函数。

1.5　递归函数

由 **1.4**，必须增加所定义的能一步步求得其值的函数。可通过增加初始函数或函数构造算子实现。

初始函数：同原始递归函数那里的。

函数构造算子：除原始递归函数那里的二个外，另加上**最小求根算子**(也称μ-算子)：设$g(x_1,\cdots,x_n,y)$是**总有定义**的已知函数，则如下定义的$f(x_1,\cdots,x_n)$为由μ-算子新定义的函数

$$f(x_1,\cdots,x_n) = \mu y[g(x_1,\cdots,x_n,y) = 0],$$

即对给定的参量$x_1,\cdots,x_n$，若有y使$g(x_1,\cdots,x_n,y)=0$时，则$f(x_1,\cdots,x_n)$取最小的这种y为值；否则，$f(x_1,\cdots,x_n)$无定义。当$f(x_1,\cdots,x_n)$不是总有定义时，我们称$f(x_1,\cdots,x_n)$是部分定义的。

那么，**递归函数集就是由上述初始函数出发，并由所给的任一函数构造算子，一步步构造出的所有的函数所成之集。**

不难看出，μ-算子可构造出部分定义的函数，而 **1.2** 中的受囿最小求根算子却只能从全定义的函数定义全定义的函数。

定理 8　(1)Arckermann 函数$A(x,y)$是全定义的递归函数。(2)原始递归函数是全定义的递归函数。

(全定义的递归函数也称一般递归函数)

证明　(2)是显然的。　(1)的证明略。

那么，是否所有可计算的函数均是递归函数呢？答案是肯定的。这从后面的研究可知。

1.6 递归函数集的分层

严格说来，将递归函数集或其子集进行分层，都属于计算复杂性的研究。或者更一般地称之为函数结构的复杂性分层研究。尚未见到将递归函数集全部进行分层的研究。但是近些年来，乔海燕、梁朴、郑锡忠及作者本人均进行过对一部分递归函数集（在原始递归函数集以上）的分层研究，且有数篇文章发表。请参考[18]。

关于计算复杂性的分层研究，也请参阅本章后面的 **3.7** 计算复杂性与 P≟NP 问题。

第二节 理想计算机

本章一开始，曾以计算机为例提出问题：计算机能力到底有多大。并指出：计算机的能力是和计算机装置、计算机程序设计语言、计算机程序直接相关的。本节将以几个有关计算的实例，深一步进行研究，分析和综合其中有关要素，从而给出理想计算机的一个描述性定义。

2.1 几个计算实例

例 18 自然数的加法和乘法运算。

这句话仅仅说明该计算的功能，即它们均是对二个自然数进行运算，而结果也为自然数，且分别是给定的二个自然数之和与积。

要实现该计算，必须要有计算装置，即它们以什么为载体进行计算。比如是以掐手指、摆棍棒，还是以一张纸和一支笔，或者是以一个算盘，甚至是以一台计算机等等。

当确定了相应的计算装置后，首先就是如何表示这些自然数

（运算对象和结果），比如说所用的装置是算盘，则就以算珠及其所摆的位置来表示它们。自然，这种表示形式要是完全确定的而且又相互统一。其次是如何操作这计算装置，即是说要有一套操作该计算装置的规则。比如说当所用的是算盘时，即是指如何将算珠拨上拨下以及“进位”、“借位”等等。这种表示和规则实际上就是使用该计算装置的语言。对算盘而言，其语言与计算机程序设计语言所不同的，只是前者算盘是不懂的，也不能阅读和运行用这种语言所写的“程序”。所有这些均要靠操作算盘的人来进行。

如何在所选定的计算装置上实现自然数的加法和乘法运算？这实际上就是如何用该计算装置的语言来“书写”或“编制”实现这些运算的步骤或“程序”，即实现这些运算的法则在该计算装置上的具体表现。比如说，当所用的计算装置是算盘，且实现的又是加法运算，则只要告诉，一位相加时如何实现当位拨珠和进位拨珠，再告诉逐位类似进行，且当逐位进行完毕后，也就实现了对具体所给加数和被加数的加法运算。再强调一下，刚刚所告诉的不是在具体作加法，而是如何作加法，即作加法的法则。显然，不管将来真正要作加法的二自然数有多大，而所告诉的法则的具体表现则是**有限表示**的。

例 19 求任给二自然数的最大公约数。

功能：对二给定的自然数进行运算，结果也为一个自然数，即所给二自然数的最大公约数。

选定实现该计算的计算装置：一张纸和一支笔。那么，表示自然数就是纸上写出的一横行连续数字，而操作该计算装置的规则就是如何在纸上写数字、划横杠和竖杠及各种记号。这就是该计算装置的语言。

用纸和笔如何计算二自然数的最大公约数，这即是常用的所谓辗转相除法，亦即实现该计算的法则（在该计算装置上的具体表现）：在纸上写下二自然数，比较它们的大小，若相等，即任意一个即为所求；否则，将较大的数除以较小的数，后用所得余数

再去除上一次的除数等等（自然还要告诉在纸上用笔如何作二数的除法）。这样直到第1次整除为止，则此时的除数即为所求。计算结束。这就是计算二数最大公约数的"程序"。当然，纸和笔也如算盘一样，它并不懂得这些程序，而是要操笔者阅读和运行该程序。

若选定实现该计算的计算装置是计算机，则其语言就是该计算机的程序设计语言。使用此语言即可将辗转相除法具体表现出来，此即实现该计算的程序。而该程序不只人能理解，计算机也能"理解"并执行它。凡是编过计算机程序的人，都知道这程序是有限表示的。然后，对任给的二自然数，运行该程序，即可得到相应的计算结果。

由上述各例可以看到：所谓一种计算装置的能力，是指用该计算装置的语言，所能编制的种种计算程序的能力。

2.2 计算的分析和理想计算机

首先要补充说明的，是2.1所举例子中，其运算对象均是离散的，且所选的计算装置对运算对象的表示也是离散的。然而，现实的计算并不都是如此，即有离散的，也有连续的；而且，现实的计算装置，其对运算对象的表示，也有连续的，如计算尺和模拟计算机即是连续的。当然，连续的计算也可选用离散的计算装置，这就要求首先要对连续的运算对象进行离散化处理，且相应的计算法则也要离散化；事实上，计算数学的很多工作都是在做这种事情。不过，本章中所讨论的均是离散的计算和离散的计算装置。

在2.1中所给的例子，显然均是可以计算的。事实上，在1.5中定义的递归函数中，除部分定义的外，均是可以计算的。那么，是否有不可计算的函数呢？当然有。从以后的研究中可以知道：

(1) 不仅有部分定义的不可计算的函数；而且有全定义的不可计算函数。

(2) 不过，在同构意义上讲，所有可计算的函数就是1.5中

定义的全定义递归函数。

现在，对 2.1 所给的计算装置作一番分析，并引出我们所要的结论。不管是算盘也好、纸和笔也好，或者是计算机也好，这些都是看得见、摸得着的，正因为如此，世界上任何现有的或将来可能造出的计算装置均为有限物，就这个意义上讲，用它们来计算任意给定多位的乘法都不可能。不过，又可以将它们作成任意有限物，比如算盘珠、杆可为任意有限多个，纸可以有任意有限多张，笔可以有任意有限支，甚至计算机也可作成相当之大，可将其字长、存贮都作得非常非常之大。因此，可假定理想计算机是潜无限的。再加上“离散”这一条件，即可假设理想计算机是任意可数、潜无限的。

现在研究一下计算装置的语言。**第一，它们对运算对象和结果的表示可假设成任意可数、潜无限的。第二，对计算装置的操作均是即刻可以完成的。**如拨一个或几个算珠，用笔在纸上写个字、划个杠，计算机执行一条指令或一个语句，都是很简单、即刻可实现的事情。

最后研究任一计算的程序。这种程序均是由计算装置的语言成分组成的，即要么是对运算对象的表示，要么是对计算装置进行操作的表示。而且任何程序又都是有限的。从计算法则的确定性，即一步步的计算法则的严格规定性，可知，**程序也是一步步地确定的**，加之语言中操作的即刻可实现性，故一步步**的程序也是即刻可实现的。**

综上所述，我们可以说，凡满足下述四条的均可以说是理想的计算，或理想计算机：

（1）计算对象是离散的，计算结果也是离散的；

（2）该计算的表示或者程序是有穷的；

（3）该计算的步骤或者说程序的运行是确定的；

（4）该计算的一步或者一步步的程序是即刻可实现的。

在本章最后，我们定义了一般的理想计算机 MMCM 之后，指出：任何一个理想的计算机即满足上述四条的，均是一MMCM。

第三节　Turing 机器

Turing 机器简称作 Turing 机，是一种特殊的理想计算机。本节先介绍 Turing 机的定义，Turing 机的功能及它与递归函数的关系。然后介绍通用 Turing 机和不可计算的函数，最后引入 Church-Turing 论题并分析 Turing 机的特殊性。顺便介绍 Turing 机的子类——有穷自动机，而这正是现代的计算机，并证明有穷自动机所计算的函数均属于 Grzegorczyk 函数类 ε^1。

3.1　Turing 机定义

（1）Turing 机的计算装置分成三部分：一条分成方格的双方向（左、右）无穷长的带子，一个时刻注视着带上某方格的读写头，一个和读写头相连接的记忆状态的寄存器（一位）。

（2）Turing 机的语言：

设 $A=\{a_1, a_2, \cdots, a_m\}$，称作符号的有穷集，且对给定的 Turing 机言，m 给定；Turing 机即用 A 中的符号，和 A 中符号的有穷串（将它们通统记作 A^*）来表示运算对象和计算结果。Turing机带子的每个方格中都可存贮一个符号。

设 $Q=\{q_1, \cdots, q_n\}$，称作状态的有穷集，且对给定的 Turing 机言，n 也是给定的；每个状态符号均可存贮在 Turing 机计算装置的记忆状态寄存器中，其表示当前 Turing 机所处的状态。

设 $M=\{l, r, h\}$，称作 Turing 机的移动符号集，l 表示 Turing 机读写头往左移一格，r 表示右移一格，h 表示不移动。

任一个五元组 $I\in Q\times A\times A\times M\times Q$，即称 Turing 机的一条指令，其是用来操作 Turing 机的计算装置的。设 $I=\{q_i, a_j, a', l, q'\}$，则当寄存器中内容为 q_i，且读写头注视格中符号为 a_j 时，即将注视格中内容改写为 a'，后左移一格，且寄存器中内容也相应改为 q'；而当上述指令中的移动符号为 r 时，即为“右移一格”，其他相同；而当动作符号为 h 时，即为“不动”。而当寄存器中内

容或方格中内容与指令中的不相符时，则 Turing 机自动停止动作，称停机。为方便起见，还经常假设 Q 中有一停机状态（比如说是 q_n），一旦寄存器中内容为 q_n 时，即停机。

（3）Turing 机程序即任一给定的指令有穷集。

（4）程序运行：为使程序运行时完全确定，总假设 Turing 机有一初始状态，比如说 q_1，在程序未运行前，q_1 先已存在寄存器中；还假设对程序的任二条指令，其前二元素不完全符合，即不会均为（q_i，a_j，…）。

设已有程序 $P=\{I_1, \cdots, I_k\}$，且设程序的运算对象已在带子上放好，且读写头总对应着（注视着）一串符号 $\alpha\in A^*$ 的最左一个符号；而未放符号的方格中恒假设均有一空白符号，比如说是 A 中的 a_m。这样，运行的准备工作都全做好。

首先从 P 中选一条指令，其第 1 个元素为 q_1，第二个元素与读写头注视的符号相同（若无这种指令时，即停机），Turing 机即按（2）中所说执行该指令，然后再类似地从 P 中选指令执行，直到停机出现。可以规定，此时从读写头注视格开始及其右边即为该程序 P 对所给的运算对象运算的结果。而如果不出现停机，则称 P 对所给运算对象无定义。

由 2.1 最末一段所说，所谓 Turing 机的能力，即是指用 Turing机语言所编的种种程序的计算能力。

如不作说明，我们恒假定 Turing 机的符号集

$$A=\{1,0\},$$

其中“0”表示空白符号。

例 20 Turing 机 M_S：其 Q $\{q_1, q_2, q_f\}$，程序为

$$\begin{array}{ccccc} q_1 & 1 & 1 & r & q_1 \\ q_1 & 0 & 1 & l & q_2 \\ q_2 & 1 & 1 & l & q_2 \\ q_2 & 0 & 0 & r & q_f \end{array}$$

不难知道，M_S 对任何一串“1”作用，结果为：在一串“1”的最

右加上一个"1"。记为 $0\underbrace{\overset{q_1}{1}\cdots1}_{x}0 \overset{M_s}{\Longrightarrow} 0\underbrace{\overset{q_f}{1}\cdots1}_{x}1$。

例 21 Turing 机 M_Z：其 $Q=\{q_1, q_2, q_f\}$，程序为

q_1	1	0	r	q_1
q_1	0	0	l	q_2
q_2	0	1	h	q_f

不难知道，M_Z 对任何一串"1"作用，结果为：仅留下一个"1"。记为 $\underbrace{\overset{q_1}{1}\cdots1}_{x}0 \overset{M_z}{\Longrightarrow} \underbrace{0\cdots0}_{x-1}\overset{q_f}{1}0$。

例 22 Turing 机 $M_{U_2^3}$：其 $Q=\{q_1, q_2, q_3, q_4, q_5, q_f\}$，而其程序为：

q_1	1	0	r	q_1
q_1	0	0	r	q_2
q_2	1	1	r	q_2
q_2	0	0	r	q_3
q_3	1	0	r	q_3
q_3	0	0	l	q_4
q_4	0	0	l	q_4
q_4	1	1	l	q_5
q_5	1	1	l	q_5
q_5	0	0	r	q_f

不难知道，$M_{U_2^3}$的作用是将初始带子上的三串"1"(以一个"0"相隔)，去掉左、右两串"1"，只留中间一串。记为 $\underbrace{\overset{q_1}{1}\cdots1}_{x_1}0\underbrace{1\cdots1}_{x_2}0\underbrace{1\cdots1}_{x_3}$ $\overset{M_{U_2^3}}{\Longrightarrow} \underbrace{0\cdots0}_{x_1}0\underbrace{\overset{q_f}{1}\cdots1}_{x_2}0\underbrace{0\cdots0}_{x_3}$。

3.2 Turing 机和递归函数

首先，研究 Turing 机所能计算的递归函数。

自然数在 Turing 机带子上的表示如下：

0，1，2，…，k，…分别表示为

1，11，111，…，$\underbrace{1\cdots1}_{k+1}$，…，记为$\underline{0}$，$\underline{1}$，$\underline{2}$，…．且假设 Turing 机 M 计算递归函数 $f(x_1，\cdots，x_n)$ 的始末过程表示为

$$0\,\underline{x_1}0\,\underline{x_2}0\cdots0\,\overset{\downarrow q_1}{\underline{x_n}}0 \overset{M}{\Rightarrow} 0\,\underline{x_1}0\,\underline{x_2}0\cdots0\,\underline{x_n}0\,\overset{\downarrow q_f}{\underline{f(x_1,\cdots,x_n)}}0,$$

其中，$\overset{\downarrow q_1}{\underline{x_n}}$ 和$\overset{\downarrow q_f}{\underline{f(x_1，\cdots，x_n)}}$分别表示 Turing 机程序运行始末的读写头注视格为$\underline{x_n}$ 和$\underline{f(x_1，\cdots，x_n)}$的最右边的“1”。而 q_1，q_f 同样表示首末状态。

那么，从**例 20**，**21**，**22** 不难做出计算递归函数 S，Z，U_i^n 的 Turing 机 $M_S{}'$，$M_Z{}'$，$M_{U_i^n}{}'$。留作练习。

例 23　Turing 机 $\mathfrak{A}$：

$$\begin{matrix} q_1 & 0 & 1 & h & q_f \\ q_1 & 1 & 1 & r & q_1 \end{matrix}，那么，\overset{\downarrow q_1}{\underline{x}}0 \overset{\mathfrak{A}}{\Rightarrow} \underline{x}\overset{\downarrow q_f}{1}。$$

现在研究一下 Turing 机的**组合**。

称 Turing 机带子上任两串“1”之间为一“**沟**”，如果它们之间的“空白”多于一个的话。

称 A 为 B，C 的**连接**，如果 A 恰为将 B 的末状态改为 C 的始状态后合在一起所构成的 Turing 机，且设 B，C 原来无相同的状态。记为 $A=BC$。那么，显然有 $A(BC)=(AB)C$。

称 A 为 B，C，D 的**分枝连接**，如果 B 的状态 q_1'，q_2' 分别改为 C，D 的首状态后而将 B，C，D 合在一起后，恰为 A，同样，设原来 B，C，D 并无相同状态。记为 $A=B\begin{cases}C\\D\end{cases}$。

称 A 为 B，C，D 的**分枝循环**，如果有 $A_1=B\begin{cases}C\\D\end{cases}$，而 A 又恰是将 A_1 中 C 的末状态改为 B 的首状态而成。记为 $A=\dot{B}\begin{cases}\dot{C}\\D\end{cases}$。

例 24　为了节省纸面，将 Turing 机 $\mathfrak{A}$：

$$\begin{matrix} q_1 & 1 & 1 & r & q_1 \\ q_1 & 0 & 1 & h & q_f \end{matrix}$$

改写为 $\begin{matrix} & 0 & 1 \\ q_1 & 1hq_f & 1rq_1 \end{matrix}$， 其功能为$\underbrace{\overset{\downarrow q_1}{1}\cdots 1}_{x}0 \overset{\mathfrak{A}}{\Rightarrow} \underbrace{1\cdots 1}_{x}\overset{\downarrow q_f}{1}$。

例 25 $\mathfrak{B}_1$：其功能如右下。

$$\begin{matrix} & 0 & 1 \\ q_1 & — & 0lq_2 \\ q_2 & 1lq_3 & 1lq_2 \\ q_3 & 0rq_1' & 1rq_2' \end{matrix}$$

$$1\underbrace{0\cdots 0}_{x}0\underbrace{1\cdots 1}_{y}\overset{\downarrow q_1}{0} \overset{\mathfrak{B}_1}{\Rightarrow} 1\underbrace{0\cdots 0}_{x}\underbrace{\overset{\downarrow q_1'}{1}\cdots 1}_{y}00。$$

$$10\underbrace{1\cdots 1}_{y}\overset{\downarrow q_1}{0} \overset{\mathfrak{B}_1}{\Rightarrow} 1\underbrace{\overset{\downarrow q_2'}{1}\cdots 1}_{y}00。$$

例 26 $\mathfrak{B}_2$：

$\begin{matrix} & 0 & 1 \\ q_1(q_1') & 0lq_f & 1rq_1 \end{matrix}$， 其功能为$\underbrace{\overset{\downarrow q_1}{1}\cdots 1}_{y}0 \overset{\mathfrak{B}_2}{\Rightarrow} \underbrace{1\cdots 1}_{y}\overset{\downarrow q_f}{}0$。

例 27 $\mathfrak{B}_3$：

$\begin{matrix} & 0 & 1 \\ q_1(q_2') & — & 0rq_f \end{matrix}$， 其功能为$\underbrace{\overset{\downarrow q_1}{1}1\cdots 1}_{y} \overset{\mathfrak{B}_3}{\Rightarrow} 0\underbrace{\overset{\downarrow q_f}{1}\cdots 1}_{y-1}$。

那么，不难知道，$\mathfrak{B}=\dot{\mathfrak{B}}_1\begin{cases}\dot{\mathfrak{B}}_2 \\ \mathfrak{B}_3\mathfrak{A}\end{cases}$ 的功能恰为：消除一个沟，即：

$$1\underbrace{0\cdots 0}_{x}\underbrace{1\cdots 1}_{y}\overset{\downarrow q_1}{0} \overset{\mathfrak{B}}{\Rightarrow} 10\underbrace{1\cdots 1}_{y}\overset{\downarrow q_f}{}\underbrace{0\cdots 0}_{x}。$$

例 28 $\mathfrak{C}$：功能为$\underbrace{1\cdots 1}_{x}\overset{\downarrow q_1}{0}0 \overset{\mathfrak{C}}{\Rightarrow} \underbrace{1\cdots 1}_{x}0\overset{\downarrow q_f}{1}$。

例 29 $\mathfrak{D}$：功能为$0\underbrace{1\cdots 1}_{x_1}0\underbrace{1\cdots 1}_{x_2}0\underbrace{1\cdots 1}_{x_3}\overset{\downarrow q_1}{0} \overset{\mathfrak{D}^2}{\Longrightarrow} 0\underbrace{1\cdots 1}_{x_1}\overset{\downarrow q_f}{0}\underbrace{1\cdots 1}_{x_2}0$ $\underbrace{1\cdots 1}_{x_3}0$（其中，$\mathfrak{D}^2$ 即 $\mathfrak{D}\mathfrak{D}$；以后，A^m 即$\underbrace{A\cdots A}_{m}$）。

例 30 $\mathfrak{E}$：功能为，判断所注视的数（此时，在最右边的"1"）为 0 或大于 0，分别转两个状态 q_1' 或 q_2'。

例 31 $\mathfrak{F}$：功能为 $0\underbrace{1\cdots1\overset{\downarrow q_1}{1}}_{x}\overset{\mathfrak{F}}{\Longrightarrow}0\underbrace{1\cdots1}_{x}\overset{\downarrow q_f}{0}$。

例 32 $\mathfrak{G}$：功能正与 $\mathfrak{D}$"相反"，即

$$0\,\underline{\overset{\downarrow q_1}{x_1}}0\,\underline{x_2}\,0\,\underline{x_3}0\overset{\mathfrak{G}^2}{\Longrightarrow}0\,\underline{x_1}0\,\underline{x_2}\,0\,\underline{x_3}\overset{\downarrow q_f}{0}。$$

例 33 $\mathfrak{H}$：功能为 $0\,x_1\underbrace{\overset{\downarrow q_1}{0}\cdots0}_{y}\,0\,x_2\overset{\mathfrak{H}}{\Longrightarrow}0\,\underline{x_1+y}\overset{\downarrow q_f}{0}x_2$。

例 34 $\mathfrak{J}_m$：$=\mathfrak{C}\dot{\mathfrak{D}}^m\mathfrak{E}\begin{cases}\mathfrak{H}\mathfrak{G}^m\\ \mathfrak{F}\mathfrak{G}^m\dot{\mathfrak{A}}\end{cases}$，其功能为

$$0\,\underline{x_1},0\,\underline{x_2}0\cdots\ 0\,\underline{\overset{\downarrow q_1}{x_m}}\ 0\,\underbrace{0\cdots0}_{x_1}\,00\overset{\mathfrak{J}_m}{\Longrightarrow}0\,\underline{x_1}0\cdots0\,\underline{x_m}0\,\underline{\overset{\downarrow q_f}{x_1}}0。$$

这里，**例 28，29，31，…，33** 均不再编制，有兴趣的读者可试构造之。而**例 30** 的 $\mathfrak{E}$ 如下：

	0	1
q_1	—	$1lq_2$
q_2	$0rq_1'$	$1rq_2'$ 。

但请读者举例运行程序 $\mathfrak{J}_3$。留作练习。

例 35 $\mathfrak{K}_m=\mathfrak{A}\,\mathfrak{J}_m^m\,\mathfrak{F}\mathfrak{D}^m\mathfrak{F}\mathfrak{G}^m$，其功能为

$$\underline{x_1}0\,\underline{x_2}0\cdots0\ \underline{\overset{\downarrow q_1}{x_m}}0\ \ \underbrace{0\cdots0}_{x_1+\cdots+x_m+2m}\overset{\mathfrak{A}}{\Longrightarrow}\underline{x_1}0\,\underline{x_2}0\cdots0\ \underline{x_m+\overset{\downarrow q_{*1}}{1}}\ \ \underbrace{0\cdots0}_{x_1+\cdots+x_m+2m}$$

$$\overset{\mathfrak{J}_m^m}{\Longrightarrow}\underline{x_1}0\,\underline{x_2}0\cdots0\,\underline{x_m+1}\,0\,\underline{x_1}0\,\underline{x_2}0\cdots0\ \underline{x_m+\overset{\downarrow q_{*2}}{1}}\,0$$

$$\overset{\mathfrak{F}}{\Longrightarrow}\underline{x_1}0\,\underline{x_2}0\cdots0\,\underline{x_m+1}0\,\underline{x_1}0\,\underline{x_2}0\cdots0\,\underline{x_m}\overset{\downarrow q_{*2}}{0}0$$

$$\overset{\mathfrak{D}^m}{\Longrightarrow}\underline{x_1}0\,\underline{x_2}0\cdots\ 0\,\underline{x_m+\overset{\downarrow q_{*3}}{1}}\,0\ \underline{x_1}0\,\underline{x_2}0\cdots0\,\underline{x_m}00$$

$$\overset{\mathfrak{F}}{\Longrightarrow}\underline{x_1}0\,\underline{x_2}0\cdots\ 0\,\underline{x_m}0\overset{\downarrow q_{*4}}{0}\,\underline{x_1}0\,\underline{x_2}0\cdots0\,\underline{x_m}00$$

$$\overset{\mathfrak{G}^m}{\Longrightarrow}\underline{x_1}0\,\underline{x_2}0\cdots0\,\underline{x_m}00\,\underline{x_1}0\,\underline{x_2}0\cdots0\,\underline{\overset{\downarrow q_f}{x_m}}00。$$

可见，其与 $\mathfrak{J}_m^m$ 所不同的，是在复制的 x_1，…，x_m 和原序列之间

多了一个“0”构成的沟。

现在来证：

定理 9 任何递归函数均可由 Turing 机计算。

证明 首先，递归函数的初始函数 S, Z, U_i^n 已分别由 Turing 机 M_s'，M_z'，$M_{U_i^n}'$ 计算。

现在对通过代入算子、递归算子和 μ-算子构造出的递归函数，编制计算它们的 Turing 机程序。

对代入算子：设 $g_1(x_1, \cdots, x_n)$，$\cdots$，$g_m(x_1, \cdots, x_n)$，$h(y_1, \cdots, y_m)$ 已分别由 M_{g_1}，$\cdots$，M_{g_m}，M_h 计算，则不难知道，

$$M_f = \mathfrak{R}_n M_{g_1} \mathfrak{I}_{n+1}^n M_{g_2} \cdots \mathfrak{I}_{n+1}^n M_{g_m} \mathfrak{I}_{(m-1)(n+1)+1} \mathfrak{I}_{(m-2)(n+1)+2} \cdots$$
$$\mathfrak{I}_{1\cdot(n+1)+(m-1)} \mathfrak{I}_{0\cdot(n+1)+m} M_h \mathfrak{L} \mathfrak{B},$$

则正好计算了函数

$$f(x_1, \cdots, x_n) = h(g_1(x_1, \cdots, x_n), \cdots, g_m(x_1, \cdots, x_n)),$$

其中，Turing 机 $\mathfrak{L}$ 的功能为

$$00\,\underline{x_1}0\,\underline{x_2}0\cdots0\,\overset{q_1\downarrow}{\underline{x_{n+1}}}0 \overset{\mathfrak{L}}{\Longrightarrow} 00\,\underbrace{0\cdots\cdots\cdots\cdots0}_{x_1+\cdots+x_n+2n}\,\overset{\downarrow q_f}{\underline{x_{n+1}}}0,$$

即除了其所注视的“数”保留外，其他，直至其左边最近一个沟间的“1”均改为“0”。

M_f 计算 $f(x_1, \cdots, x_n)$ 的过程为

$$,\underline{x_1},\cdots,\overset{q_1\downarrow}{\underline{x_n}},\langle 经\ \mathfrak{R}_n\rangle,\underline{x_1},\cdots,\underline{x_n},,\underline{x_1},\cdots,\overset{\downarrow}{\underline{x_n}},$$

$$\langle 经\ M_{g_1}\rangle,\underline{x_1},\cdots,\underline{x_n},,\underline{x_1},\cdots,\underline{x_n},\overset{\downarrow}{\underline{g_1(x_1,\cdots,x_n)}},$$

$$\langle 经\ \mathfrak{I}_{n+1}^n\rangle,\underline{x_1},\cdots,\underline{x_n},,\underline{x_1},\cdots,\underline{x_n},\underline{g_1(x_1,\cdots,x_n)},\underline{x_1},\cdots,\overset{\downarrow}{\underline{x_n}},$$

$$\langle 经\ M_{g_2}\rangle,\underline{x_1},\cdots,\underline{x_1},,\underline{x_1},\cdots,\underline{x_n},\underline{g_1(x_1,\cdots,x_n)},\underline{x_1},\cdots,\underline{x_n},$$
$$\overset{\downarrow}{\underline{g_2(x_1,\cdots,x_n)}},$$

$\cdots\cdots$

$$\langle 经\ M_{g_m}\rangle,\underline{x_1},\cdots,\underline{x_n},,\underline{x_1},\cdots,\underline{x_n},\underline{g_1},\underline{x_1},\cdots,\underline{x_n},\underline{g_2},\cdots,\underline{x_1},\cdots,$$
$$\underline{x_n},\overset{\downarrow}{\underline{g_m(x_1,\cdots,x_n)}},$$

$\langle$经 $\mathfrak{I}_{(m-1)(n+1)+1}\rangle, \underline{x_1}, \cdots, \underline{x_n}, , \underline{x_1}, \cdots, \underline{g_m}, \underline{g_1(x_1, \cdots, x_n)}$,

$\langle$经 $\mathfrak{I}_{(m-2)(n+1)+2}\rangle, \underline{x_1}, \cdots, \underline{x_n}, , \underline{x_1}, \cdots, \underline{g_m}, \underline{g_1(x_1, \cdots, x_n)}$,

$g_2\ \underline{(x_1, \cdots, x_n)}$,

⋮

$\langle$经 $\mathfrak{I}_{0(n+1)+m}\rangle, \underline{x_1}, \cdots, \underline{x_n}, , \underline{x_1}, \cdots, \underline{g_m}, \underline{g_1(x_1, \cdots, x_n)}, \cdots$,

$\underline{g_m(x_1, \cdots, x_n)}$,

$\langle$经 $M_h\rangle, \underline{x_1}, \cdots, \underline{x_n}, , \underline{x_1}, \cdots, \underline{g_m}, \underline{g_1}, \cdots, \underline{g_m}, \underline{h(g_1, \cdots, g_m)}$,

$\langle$经 $\mathfrak{L}\rangle, \underline{x_1}, \cdots, \underline{x_n}, 0, \cdots, 0, \underline{h(g_1, \cdots, g_m)}$,

$\langle$经 $\mathfrak{B}\rangle, \underline{x_1}, \cdots, \underline{x_n},\ \underline{h(g_1, \cdots, g_m)}^{q_f}$。

对递归算子：设 $g(x_1, \cdots, x_n)$、$h(y_1, \cdots, y_{n+2})$分别由 Turing 机 M_g、M_h 计算，则不难知道，

$$M_f = \mathfrak{R}_{n+1} M_g \mathfrak{I}_{n+2} \mathfrak{S}\begin{cases}\mathfrak{I}_2 \mathfrak{L}\ \mathfrak{B} \\ \mathfrak{C}\ \mathfrak{I}_3 \mathfrak{I}^n_{n+4}\end{cases} M_h \mathfrak{I}_{n+4} \mathfrak{F}\ \mathfrak{S}\begin{cases}\mathfrak{I}_2\ \mathfrak{L}\ \mathfrak{B} \\ \mathfrak{I}_{n+4}\ \mathfrak{A},\end{cases}$$

正好计算了函数 $f(y, x_1, \cdots, x_n)$，其定义为

$$\begin{cases} f(0, x_1, \cdots, x_n) = g(x_1, \cdots, x_n) \\ f(y', x_1, \cdots, x_n) = h(y, f(y, x_1, \cdots, x_n), x_1, \cdots, x_n)。\end{cases}$$

其中，$\mathfrak{L}$ 即在证对代入算子时所构造的。

M_f 计算 $f(y, x_1, \cdots, x_n)$的过程为：(带上内容)

$, y, x_1, \cdots, x_n, , y, x_1, \cdots, x_n, g(x_1, \cdots, x_n)$,

$$y, \begin{cases} g(x_1, \cdots, x_n), \text{(后边即为 } \mathfrak{L}\ \mathfrak{B} \text{ 运行结果)}, & \text{如果 } y = 0; \\ 0, g(x_1, \cdots, x_n), x_1, \cdots, x_n, h(0, g(x_1, \cdots, x_n), \\ \qquad x_1, \cdots, x_n), & \text{如果 } y \neq 0; \end{cases}$$

$$y - 1\begin{cases} h(0, g(x_1, \cdots, x_n), x_1, \cdots, x_n), \\ \quad \text{(后边即为 } \mathfrak{L}\ \mathfrak{B} \cdots), & \text{如果 } y - 1 = 0; \\ 1, h(0, g(x_1, \cdots, x_n), x_1, \cdots, x_n), x_1, \cdots, x_n, h(1, h(0, g \\ \quad (x_1, \cdots, x_n), x_1, \cdots, x_n), x_1, \cdots, x_n), & \text{如果 } y - 1 \neq 0; \end{cases}$$

$$y - 2\begin{cases} h(1, h(0, g(x_1, \cdots, x_n), x_1, \cdots, x_n), x_1, \cdots, x_n), \\ \quad \text{(后边} \cdots), & \text{如果 } y - 2 = 0; \\ 2, \cdots & \text{如果 } y - 2 \neq 0; \end{cases}$$

对 μ-算子：类似对递归算子的证明，只是更为简单些。这里略。至此，**定理 9** 证毕。

现在研究另一面，即任给的 Turing 机 T_M，其所计算的函数均为递归函数。

为确定起见，不妨令

$$A = \{a_0, a_1, \cdots, a_{m_1}\},$$

$$Q = \{q_0, q_1, \cdots, q_{m_2}\},$$

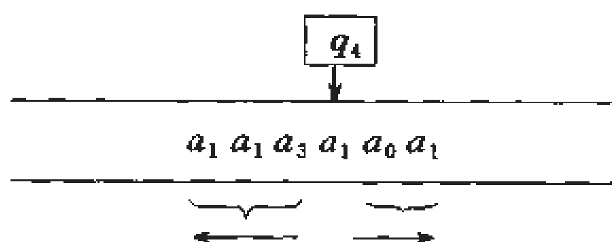
且设前述的"0"(表示"空白")、"1"(即一进制数字)分别为"a_0"、"a_1"，而停机状态为 q_0。另外，对可能造成"自然停机"的 Turing 机 T_M 的程序 P 进行修改，使其同原来的功能一样，只是碰到"自然停机"时，均改成转入停机状态"q_0"。如下修改 P：

对任给的 $q_i, a_j (i=1,\cdots,m_2;\ j=0,1,\cdots,m_1)$，若 P 中无指令 $(q_i, a_j, \cdots)$ 时，则添加一条指令 (q_i, a_j, a_j, h, q_0)。易知，这样的修改满足要求。

定理 10 任给定的 Turing 机 T_M 所计算的函数均为递归函数。

证明 分下述五步证明之。

(1)T_M **的动态信息为带上的内容和** T_M **所处的状态**，用一自然数表示之。若在某时刻，T_M 为



则表示该时刻 T_M **动态信息的自然数**(简称为**动态信息**)为

$$w = 2^{5^1 3^1 2^3} \cdot 3^1 \cdot 5^4 \cdot 7^{2^0 3^1},$$

那么，$(w)_1 = 1$，即 T_M 注视格中符号的下标；

$(w)_2 = 4$，即 T_M 所处状态的下标；

$(w)_0 = 5^1 3^1 2^3$，即注视格左边带上的内容，且$((w)_0)_0$为注视格左邻符号的下标，$((w)_0)_1$为注视格左二符号的下标，$((w)_0)_2$为注视格左三符号的下标；

$(w)_3 = 2^0 3^1$，即注视格右边带上的内容，且$((w)_3)_0$，$((w)_3)_1$，$((w)_3)_2$，…分别为注视格右第1，第2，第3，…符号的下标。

(2)T_M的**动作函数**：设w_1为某时刻动态信息。

若令$u=(w_1)_0$，$v=(w_1)_3$，且设注视格内容为a_b，T_M处于状态q_c，则$w_1 = 2^u \cdot 3^b \cdot 5^c \cdot 7^v$；那么，当$T_M$执行其指令$(q_c, a_b, a_d, l, q_e)$后，$T_M$的动态信息则为

$$w_2 = 2^{\prod_{i<u} p_i^{(u)_{i+1}}} \cdot 3^{(u)_0} \cdot 5^e \cdot 7^{2^d \cdot \prod_{i<v} p_{i+1}^{(v)_i}},$$

这样，T_M的指令(q_c, a_b, a_d, l, q_e)即为一函数$\rho_{b,c}(u,v) = 2^{\prod_{i<u} p_i^{(u)_{i+1}}} \cdot 3^{(u)_0} \cdot 5^e \cdot 7^{2^d \cdot \prod_{i<v} p_{i+1}^{(v)_i}}$；显然，$T_M$的任指令$(q_{c_1}, a_{b_1}, \cdots)$相应的函数$\rho_{b_1,c_1}(u,v)$均是原始递归函数。令

$$\rho(w) = \begin{cases} \rho_{b,c}((w)_0,(w)_3), & \text{如果}(w)_1 = b,(w)_2 = c, \\ & \text{且有指令}(q_c,a_b,\cdots)\text{时}, \\ w, & \text{否则。} \end{cases}$$

称$\rho(w)$为T_M的**动作函数**。易知，$\rho(w)$原始递归。

(3)T_M在t时刻后的动态信息$\theta(w,t)$，可定义为

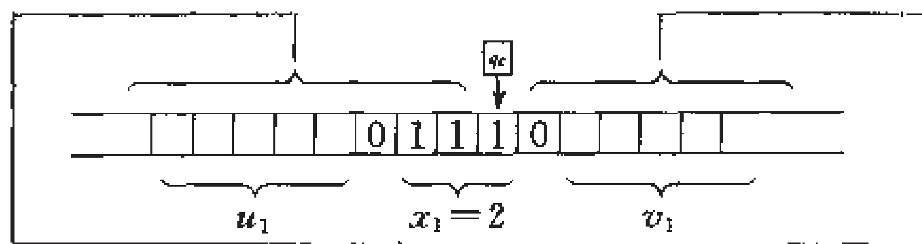
$$\begin{cases} \theta(w,0) = w \\ \theta(w,t') = \rho(\theta(w,t)), \end{cases}$$

显然，$\theta(w,t)$也原始递归。

(4)Turing 机T_M计算函数时，初始时刻的动态信息为：

$$\tau_1(x_1,c,u_1,v_1) = 2^{\prod_{i<x_1} p_i^1 \cdot p_{x_1}^0 \cdot \prod_{i<u_1} p_{x_1+i}^{(u_1)_i}} \cdot 3^1 \cdot 5^c \cdot 7^{2^0 \cdot \prod_{i<v_1} p_{i+1}^{(v_1)_i}},$$

例如：

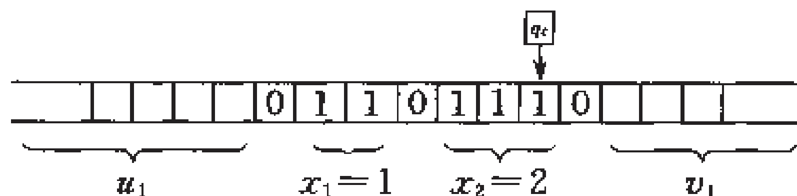


$$\tau_1(2,c,u_1,v_1) = 2^{2^1\cdot 3^1\cdot p_2^0\cdot \prod_{i<u_1} p_{3+i}^{(u_1)_i}}\cdot 3^1\cdot 5^c\cdot 7^{2^0\cdot \prod_{i<v_1} p_{i+1}^{(v_1)_i}}。$$

当然，开始时，$c=1$，$u_1=v_1=1$；但这里为使用 τ_1 定义 $\tau_{n+1}(x_1,\cdots,x_n,x_{n+1},c,u_1,v_1)$，故才有上述一般地构造。事实上，对 $n=1,2,3,\cdots$，有

$$\tau_{n+1}(x_1,\cdots,x_n,x_{n+1},c,u_1,v_1)=$$
$$\tau_1(x_{n+1},c,(\tau_n(x_1,\cdots,x_{n-1},x_n+1,c,u_1,v_1))_0,v_1),$$

例如，



∵ $\tau_1(x_1+1,c,u_1,v_1)$为在 x_1 和 x_2 间加上"1"后，再将 x_2 从带中去掉(剪掉)所得的动态信息，

∴ $(\tau_1(x_1+1,c,u_1,v_1))_0$ 恰为自刚加上的"1"之左的带上的内容，

因而，$\tau_2(x_1,x_2,c,u_1,v_1)=\tau_1(x_2,c,(\tau_1(x_1+1,c,u_1,v_1))_0,v_1)$。

当 T_M 计算一个 n 元函数 $\varphi(x_1,\cdots,x_n)$时，初始时动态信息应为 $\tau_n(x_1,\cdots,x_n,1,1,1)$，而结束时动态信息则为 $\tau_{n+1}(x_1,\cdots,x_n,x,0,u_1,v_1)$，且其中 x 恰为计算结果。

(5)由上述，若 $\varphi(x_1,\cdots,x_n)$有定义当且仅当有四元组(t,x,u_1,v_1)，使得

$$\theta(\tau_n(x_1,\cdots,x_n,1,1,1),t)=$$
$$\tau_{n+1}(x_1,\cdots,x_n,x,0,u_1,v_1),$$

且 x 即 $\varphi(x_1,\cdots,x_n)$的值。而这又当且仅当有 z，使

$$(z)_0 = t, (z)_1 = x, (z)_2 = u_1, (z)_3 = v_1,$$

且 $\theta(\tau_n(x_1, \cdots, x_n, 1, 1, 1), (z)_0) =$

$$\tau_{n+1}(x_1, \cdots, x_n, (z)_1, 0, (z)_2, (z)_3),$$

$(z)_1 = x$ 即 $\varphi(x_1, \cdots, x_n)$，故有

$$\varphi(x_1, \cdots, x_n) \simeq (\mu z[\theta(\tau_n(x_1, \cdots, x_n, 1, 1, 1), (z)_0) = \tau_{n+1}(x_1, \cdots, x_n, (z)_1, 0, (z)_2, (z)_3)])_1。$$

上述"$\simeq$"表示：其两边的函数同时有定义，且当有定义时，值又相等。

显然，$\varphi(x_1, \cdots, x_n)$是递归函数。且知：$\mu$-算子只用了一次，而其他均是原始递归的。这就完成了**定理 10** 的证明。

3.3 通用函数和递归定理

(1)通用函数

由**定理 10** 每个 Turing 机所计算的函数

$$\varphi(x_1, \cdots, x_n) \simeq (\mu z(\theta(\tau_n(x_1, \cdots, x_n, 1, 1, 1), (z)_0) = \tau_{n+1}(x_1, \cdots, x_n, (z)_1, 0, (z)_2, (z)_3)))_1,$$

其中，除 μ-算子用了一次外，其他均是原始递归函数。故也可写成

$$\varphi(x_1, \cdots, x_n) \simeq h(\mu z(g(x_1, \cdots, x_n, z) = 0)),$$

其中，$h(x) = (x)_1$，$g(x_1, \cdots, x_n, z) = |\theta(\tau_n(\quad), (z)_0) - \tau_{n+1}(\quad)|$，$g$，$h$ 均原始递归。

再由**定理 9**，即每个递归函数均可由 Turing 机计算，故任一递归函数 $\varphi(x_1, \cdots, x_n)$，可写成

$$\varphi(x_1, \cdots, x_n) = h(\mu z(g(x_1, \cdots, x_n, z) = 0)),$$

h 与 φ 无关，g 与 φ 相关，且 h，g 均原始递归。

不难构造 n-元原始递归函数的通用函数 $U'(x_1, \cdots, x_n, y)$，使对任原始递归函数 $\varphi(x_1, \cdots, x_n)$，均可找到 y_0，使

$$\varphi(x_1, \cdots, x_n) = U'(x_1, \cdots, x_n, y_0),$$

且对任自然数 k，$U'(x_1, \cdots, x_n, k)$均为原始递归函数。(事实上，可类似证明单重嵌套递归式可归约为递归式而构造 U'。)

由上述，n-元递归函数的通用函数

$$U(x_1,\cdots,x_n,y)=h(\mu z(U'(x_1,\cdots,x_n,z,y)=0)),$$

其中,U'为$n+1$元原始递归函数的通用函数。

也可如下从1-元递归函数的通用函数而通过配对函数$J(x,y)$,$K(z)$,$L(z)$构造出任n-元递归函数的通用函数。

设$U^{(2)}(x,y)$为一元递归函数的通用函数,令

$$U^{(n+1)}(x_1,\cdots,x_n,y)=U^{(2)}(J^{(n+1)}(x_1,\cdots,x_n,L(y)),K(y)),$$

其中,$J^{(2)}(x,y)=J(x,y)$,且$K(J(x,y))=x$,$L(J(x,y))=y$,

$J^{(n+1)}(x_0,\cdots,x_n)=J(x_0,J^{(n)}(x_1,\cdots,x_n))$,对$n\geqslant 2$;那么,有:

定理 11 $U^{(n+1)}(x_1,\cdots,x_n,y)$为$n$-元递归函数的通用函数,$n\geqslant 2$。

证明 该定理,可按通用函数的定义证明之,即对任给n-元递归函数$\varphi(x_1,\cdots,x_n)$,可找到k_0,使$\varphi(x_1,\cdots,x_n)=U^{(n+1)}(x_1,\cdots,x_n,k_0)$。这里也可按另种更好的方式证明,即对任给的$n$-元递归函数的通用函数$W(x_1,\cdots,x_n,k)$,均可找到$k_0$,使

$$W(x_1,\cdots,x_n,k)=U^{(n+1)}(x_1,\cdots,x_n,J(k_0,k))。$$当然,

$$W(x_1,\cdots,x_n,k)=W(C_1^{(n+1)}(J^{(n+1)}(x_1,\cdots,x_n,k)),\cdots,C_{n+1}^{(n+1)}(J^{(n+1)}(x_1,\cdots,x_n,k))),$$

只要,$C_i^{(n+1)}(J^{(n+1)}(x_1,\cdots,x_n,x_{n+1}))=x_i$,$i=1,\cdots,n+1$。事实上,可令$C_1^{(n+1)}(x)=K(x)$,$C_2^{(n+1)}(x)=K(L(x))$,$\cdots$,$C_n^{(n+1)}(x)=K(\underbrace{L(\cdots L}_{n-1}(x)\cdots))$,$C_{n+1}^{(n+1)}(x)=\underbrace{L(\cdots L}_{n}(x)\cdots)$。

因为设$g(x)=W(C_1^{(n+1)}(x),\cdots,C_{n+1}^{(n+1)}(x))$,故有$k_0$,使

$$g(x)=U^{(2)}(x,k_0),$$

那么,

$$\begin{aligned}W(x_1,\cdots,x_n,k)&=g(J^{(n+1)}(x_1,\cdots,x_n,k))\\&=U^{(2)}(J^{(n+1)}(x_1,\cdots,x_n,k),k_0)\\&=U^{(2)}(J^{(n+1)}(x_1,\cdots,x_n,L(J(k_0,k))),\\&\qquad K(J(k_0,k)))\\&=U^{(n+1)}(x_1,\cdots,x_n,J(k_0,k))。\end{aligned}$$

证毕。

在该定理证明中，也可看到：对任给的通用函数 $W(x_1,\cdots,x_n,k)$，均可构造函数 h_{k_0}，k_0 依赖于 W，使

$$W(x_1,\cdots,x_n,k)=U^{(n+1)}(x_1,\cdots,x_n,h_{k_0}(k)),$$

$h_{k_0}(k)=J(k_0,k)$。

另外，从 1 -元递归函数的通用函数 $U^{(2)}(x,k)$ 构造 n -元递归函数的通用函数，可以更自然地如下定义：

$$U_1^{(3)}(x_1,x_2,k)=U^{(2)}(J(x_1,x_2),k),$$

$$U_1^{(4)}(x_1,x_2,x_3,k)=U_1^{(3)}(J(J(x_1,x_2),x_3),k),\cdots。$$

(2)$s-m-n$ 定理

任何一个递归函数 $\varphi(x_1,\cdots,x_n,y_1,\cdots,y_m)$，既可把它看作是 $n+m$ 元函数，故可找到 k_0，使

$$\varphi(x_1,\cdots,x_n,y_1,\cdots,y_m)=U^{(n+m+1)}(x_1,\cdots,x_n,y_1,\cdots,y_m,k_0);$$

也可将 $y_1,\cdots,y_m$ 作为给定的参量，而成为 n 元函数 $\varphi(x_1,\cdots,x_n,y_1,\cdots,y_m)$，故又可找到 k_1，使

$$\varphi(x_1,\cdots,x_n,y_1,\cdots,y_m)=U^{(n+1)}(x_1,\cdots,x_n,k_1)。$$

现在即来研究 k_0,k_1 之间的关系。这就是 $s-m-n$ 定理的直观涵义。该定理为：

定理 12 用 $\varphi_k^{(n)}$ 表示由通用函数 $U^{(n+1)}$ 枚举时，枚举数为 k 的 n -元递归函数，则：对任意 $m,n\geqslant1$，有一般递归函数 $S_n^m(k,y_1,\cdots,y_m)$，使

$$\varphi_k^{(n+m)}(x_1,\cdots,x_n,y_1,\cdots,y_m)$$

$$=\varphi^{(n)}_{S_n^m(k,y_1,\cdots,y_m)}(x_1,\cdots,x_n)。$$

证明　首先证：(＊)对任何$(n+m)$元递归函数 $g(x_1,\cdots,x_n,y_1,\cdots,y_m)$，可找到一般递归函数 $h(y_1,\cdots,y_m)$，使

$$g(x_1,\cdots,x_n,y_1,\cdots,y_m)=U^{(n+1)}(x_1,\cdots,x_n,h(y_1,\cdots,y_m))。$$

1)当 $m=1$ 时，令

$$\overline{U}(x_1,\cdots,x_n,k)=\begin{cases} g\left(x_1,\cdots,x_n,\dfrac{k}{2}\right), & \text{当 } k \text{ 为偶数,} \\ U^{(n+1)}\left(x_1,\cdots,x_n,\dfrac{k-1}{2}\right), & \text{当 } k \text{ 为奇数;} \end{cases}$$

那么,因为 $\overline{U}$ 取遍了 $U^{(n+1)}$ 的一切值(当 k 为奇数时取遍了),故 $\overline{U}$ 也是通用函数。故由定理 1 的证明中,有一般递归函数 $h_{k_0}(k)$,使

$$\overline{U}(x_1,\cdots,x_n,k)=U^{(n+1)}(x_1,\cdots,x_n,h_{k_0}(k)),$$

k_0 与 $\overline{U}$ 相关。故有

$$g(x_1,\cdots,x_n,y_1)=U^{(n+1)}(x_1,\cdots,x_n,h_{k_0}(2y_1))$$

($\because k$ 是偶数时,$\overline{U}(x_1,\cdots,x_n,k)$即能取遍所有 $g(x_1,\cdots,x_n,\frac{k}{2})$,而

$$\overline{U}(x_1,\cdots,x_n,2k)=U^{(n+1)}(x_1,\cdots,x_n,h_{k_0}(2y_1))。),$$

这 $h_{k_0}(2y_1)$即为所求的 $h(y_1)$。

2)当 $m>1$ 时,因为 $g(x_1,\cdots,x_n,C_1^{(m)}(y),\cdots,C_m^{(m)}(y))$是$(n+1)$元的,故类似 1)中的证明,有 $h(y)$,使

$$g(x_1,\cdots,x_n,C_1^{(m)}(y),\cdots,C_m^{(m)}(y))=U^{(n+1)}(x_1,\cdots,x_n,h(y)),$$

取 $y=J^{(m)}(y_1,\cdots,y_m)$,则得:

$$g(x_1,\cdots,x_n,y_1,\cdots,y_m)=U^{(n+1)}(x_1,\cdots,x_n,h(J^{(m)}(y_1,\cdots,y_m)))。$$

故这里的 h 应为 $h(J^{(m)}(y_1,\cdots,y_m))$。故(*)证毕。

对任何$(n+m)$元递归函数

$$\varphi(x_1,\cdots,x_n,y_1,\cdots,y_m),$$

可表示为 $U^{(n+m+1)}(x_1,\cdots,x_n,y_1,\cdots,y_m,k)$,对此 $U^{(n+m+1)}$,由(*),可找到一般递归函数 $h(k,y_1,\cdots,y_m)$,使

$$\begin{aligned} &U^{(n+m+1)}(x_1,\cdots,x_n,y_1,\cdots,y_m,k) \\ &\quad=U^{(n+1)}(x_1,\cdots,x_n,h(k,y_1,\cdots,y_m)), \end{aligned}$$

这 h 与 $U^{(n+m+1)}$有关,而与 φ 无关。

故 φ 作为$(n+m)$元函数的号码(即枚举数)为 k,而作为 n 元函数的号码为 $h(k,y_1,\cdots,y_m)$,故得**定理 12** 的证明,而 $h(k,y_1,\cdots,y_m)$即定理中的 $S_n^m(k,y_1,\cdots,y_m)$。证毕。

(3)递归定理

定理 13(递归定理) 对任一递归函数 $h(x)$,可找到常数 a,使 $\varphi_a=\varphi_{h(a)}$,其中,φ_k 表示由通用函数 U 枚举时,枚举数为 k 的递归函数。

现在简单说一下该定理的涵义。其即是说,在通用函数 U 的枚举下,对任给 h,总会有一个不动点 a。

证明**定理 13**,只需对 $U^{(2)}$ 进行证明。

由 $s-m-n$ 定理,有 $S_1^1(k,y)$,使

$$U^{(3)}(x,y,k)=U^{(2)}(x,S_1^1(k,y)),$$

那么,有 p,使

$$\begin{aligned}U^{(2)}(x,h(S_1^1(y,y)))&=U^{(3)}(x,y,p)\\&=U^{(2)}(x,S_1^1(p,y)),\end{aligned}$$

故

$$U^{(2)}(x,h(S_1^1(p,p)))=U^{(2)}(x,S_1^1(p,p)),$$

因而,$a=S_1^1(p,p)$。证毕。

3.4 通用 Turing 机

从本节研究可知,Turing 机程序是不存入机器的带子上的,而是由人在按指令的规定,一步步执行的。这和现在的计算机是不相同的。那么,能否定义一种 Turing 机,使其可执行存入带子上的程序呢? 回答是肯定的。

设任给定符号的有穷集 A、C,且 $A\subseteq C$。称函数 $\ulcorner\ \urcorner:C^*\rightarrow A^*$ 为一**编码函数**,如果满足:(1) $\ulcorner\ \urcorner$ 为全定义的 Turing 机可计算函数(所谓全定义即对任给 $c\in C^*$,该 Turing 机总会停机);(2) $\ulcorner\ \urcorner$ 为 1-1 的,即对任 $\alpha_1,\alpha_2\in C^*$,若 $\ulcorner\alpha_1\urcorner=\ulcorner\alpha_2\urcorner$,则 $\alpha_1=\alpha_2$。此时,称 $\ulcorner\alpha\urcorner$ 为 α 的编码。

事实上,在本章 3.2 中自然数在 Turing 机带子上的表示也是一种编码。

所谓**通用 Turing 机 T**,实际上也是一个 Turing 机,只不过其

功能较强。可以使任给的 Turing 机 T'，若 $\ulcorner T' \urcorner$ 为 T' 的编码，则满足

$$T'': 0\,\underbrace{\overset{\downarrow q_0}{1}\cdots 1}_{x}0 \Longrightarrow 0\,\underbrace{\overset{\downarrow q_f}{1}\cdots 1}_{y}0$$

当且仅当

$$T: *\ \ulcorner T' \urcorner\ *\ \underbrace{\overset{\downarrow q_0}{1}\cdots 1}_{x}0 \Longrightarrow *\ \ulcorner T'' \urcorner\ *\ \underbrace{\overset{\downarrow q_f}{1}\cdots 1}_{y}。$$

这里自然包含：T''停机当且仅当 T 停机，而当停机时，结果如上。

现在从 T 的功能出发，介绍一下如何构造 T。为确定起见，我们设任给定的 Turing 机，其输入符号集均为 $A=\{0,1\}$；为了方便构造，而设 T 的输入符号集 $A_T=\{0,1,*\}$。且设 T''不会出现自然停机。

(1)T 的功能：T 模拟任给定的 T'的功能。具体说来，T 首先记下其读写头位置(相应于 T'的位置)，而后左移至 T'的编码部分，去找出开始状态编码，根据该指令的功能，再右移出编码区，找到刚才记下的位置，实现该指令的作用；而后，又记下读写头的新位置，再左移至编码区，找到有关的指令去执行…，依此类推。一当所找的指令要转入最后状态，则在实现了该指令的作用后，即刻转入停机。

(2)任给 T'的编码 $\ulcorner T' \urcorner$：

a_i 的编码为 $\underbrace{1\cdots 1}_{i+1}$

q_j 的编码为 $\underbrace{1\cdots 1}_{j+1}$

l 的编码为 1

r 的编码为 11

h 的编码为 111

那么，T'的指令 $I=(q_i,a_j,a_k,v,q_p)$($v=l$ 或 r 或 h)的编码为

$$\ulcorner q_i \urcorner\ 0\ \ulcorner a_j \urcorner\ 0\ \ulcorner a_k \urcorner\ 0\ \ulcorner v \urcorner\ 0\ \ulcorner q_p \urcorner\ ,$$

而 T' 的程序$\{I_1,\cdots,I_m\}$(不妨设已按"q_i"、"a_j"的下标大小的字典序排好)的编码则为

$$110 \ulcorner I_1 \urcorner 0 \ulcorner I_2 \urcorner 0 \ulcorner I_3 \urcorner 0 \cdots 0 \ulcorner I_m \urcorner ,$$

其中，最左的“11”表示开始状态“q_1”。

(3)几个构造技巧：

为要记下 T 的读写头位置(实际上是相应于 T' 的读写头位置)，可在该格子中写为“ * ”，而当指令处理完后，必须将此格中内容恢复，这就要求在写“ * ”之前，根据是“0”或“1”而转入不同的状态记忆之。

上述(2)中最左的安排，实际上是标志当前状态的，以“ * ”标记在该状态之“左”邻。这便要求，当处理到哪个状态时，在往右移出编码区，去实现当前指令的作用之前，必须在该指令要转入的状态编码之左标记“ * ”，以便下一条指令的执行。当然，指令执行完后，找到了新的指令，还要把上次标记的状态之左的“ * ”重新恢复为“0”。

一条指令处理完后，就要找下条应执行的指令。而这已在上条指令的要转入状态编码之左标记有“ * ”，根据此状态，来寻找下条应执行的指令。因而，首先要将此状态，与各指令的状态进行“比较”，其次还要“比较”指令中的注视格中符号与当前“T'”的注视格中符号(而这已通过状态记忆之)。前者可类似**定理 8** 中 $\mathfrak{J}_m$ 的构造；后者因已记在状态之中，故变得十分简单了。当然，因为我们限定了任给的 Turing 机 T'，其输入符号集固定，即 $A=\{0,1\}$，故可采取这种将所注视的符号“记忆”在状态之中；而若 A 可为任意有穷集时，则不能这样作了，只能类似“比较”状态的办法进行处理。

在作了上述规定和注释后，这里就不再直接构造通用 Turing 机了。请读者补充构造之，这对读者掌握 Turing 机的构造方法和技巧是大有好处的。

从通用 Turing 机的功能和运行不难看到，其是将任给的 Turing 机 T' 的程序存入到带子上的，这与现在的一般计算机很相象。事实上，1945 年第 1 台计算机的设计者之一，von Neumann 就说过，他设计那样的计算机是受了通用 Turing 机器理论的影响。当然，通用 Turing 机自己也是个一般的 Turing 机，其程序也是存

在带子之外，而要靠人在执行其程序的。从这个意义上说，通用 Turing 机不过是个解释程序而已，且这个解释程序又是靠人在执行的。

3.5 通过 Turing 机定义的非递归函数

现在研究全定义的非递归函数。

称 Turing 机 T 为**打印机**，如果其输入符号集 $A=\{0,1\}$，且功能为 $\overset{\downarrow q_1}{0} \overset{T}{\Longrightarrow} \overset{\downarrow q_f}{0}\underbrace{1\cdots1}_{i}$，即从空白带出发，打印一系列的"1"，且带上除这些"1"外，余处全为"0"。记为 $T(0)=i$。

例 36 $T^{(4)}$为四个状态的打印机(忽略停机状态)，可如下造之：

$$
\begin{array}{ccccc}
q_1 & 0 & 1 & r & q_2 \\
q_2 & 0 & 1 & r & q_3 \\
q_3 & 0 & 1 & l & q_4 \\
q_4 & 1 & 1 & l & q_4 \\
q_4 & 0 & 1 & h & q_f
\end{array}
$$
。

从这个例子可知，易造出具有$(x+1)$个状态的打印机，而其打印出$\underbrace{1\cdots1}_{x}$，记为$1^{(x)}$。

令 $\Sigma(x)=\max l\{T^{(x+1)}(0)=l \mid T^{(x+1)}$为 $x+1$个状态的打印机$\}$。

那么，该 $\Sigma(x)$是确定的。因为，对任给的 x_0，且有 x_0+1个状态的打印机只有有穷个，故 $\Sigma(x_0)$有有限值。现在已知：

$$\Sigma(1)=1,$$

$$\Sigma(2)=4,$$

$$\Sigma(3)=6,$$

$$\Sigma(4)=13,$$

$$\Sigma(5) \geqslant 501^{1)}。$$

有下述定理：

定理14 $\Sigma(x)$非递归函数。

证明　只要能证明，对任给的递归函数 $f(x)$，均有 x_0，使 $\Sigma(x_0) > f(x_0)$，即毕。现证明如下。

令 $g(x) = \sum_{i=0}^{x}[f(i)+i^2]$，故 $g(x)$自然也为递归函数，且不难证明，对任给的 x，均有

(1)$g(x) \geqslant f(x)$，

(2)$g(x) \geqslant x^2$，

(3)$g(x+1) > g(x)$。

那么，由**定理9**，不难构造计算 $g(x)$的 Turing 机 T_g。即有

$$\overset{\downarrow q_1}{0}1^{(x+1)}0 \overset{T_g}{\Longrightarrow} \overset{\downarrow q_f}{0}1^{(g(x)+1)}0。$$

不妨设 T_g 有 c 个状态。

现在构造 Turing 机

$$T_x = T^{(x+1)} \cdot T_g^2,$$

那么，

$$\overset{q_1 \downarrow}{0} \overset{T^{(x+1)}}{\Longrightarrow} \overset{q_{f_1}\downarrow}{0}1^{(x+1)}0 \overset{T_g}{\Longrightarrow} \overset{q_{f_2}\downarrow}{0}1^{(g(x)+1)}0 \overset{T_g}{\Longrightarrow} \overset{q_f\downarrow}{0}1^{(g(g(x))+1)}0;$$

由**例36**，不妨设 $T^{(x+1)}$的状态数为 $x+2$。故 T_x 的状态数为 $x+2c+1$。因而，T_x 可看成有 $x+2c+1$个状态的打印机。故由 Σ 函数定义，有：

(4)$\Sigma(x+2c+1) \geqslant g(g(x))+1$。

易知，当 x 足够大时，$x^2 > x+2c+1$；再结合上述(2)，可得：当 x 足够大时，

(5)$g(x) > x+2c+1$；

再由 $g(x)$的单调递增性，可得：

(6)$g(g(x))+1 > g(x+2c+1)+1$；

---

1)　请见本节后面，现已知 $\Sigma(5) \geqslant 4096$。

再由(1),又可得:

(7) $g(x+2c+1) \geqslant f(x+2c+1)$;

这样,由(4),(6),(7),可得:当 x 足够大时,

$$\Sigma(x+2c+1) > f(x+2c+1)。$$

这就是说,Σ 不可能是任何递归函数 f。证毕**定理14**。

上述 Σ 函数被称作**"忙海狸"**。1983年,在联邦德国举行了一次世界性的忙海狸竞赛。凡参加者均各自构造5状态的打印机,谁构造的打印1的个数最多时,谁即为优胜者。结果,优胜者是参予评判忙海狸的人,他构造的打印机可打印501个"1";而第二名构造的打印机只能打印240个"1";即 $\Sigma_1(5)=501$,$\Sigma_2(5)=240$,且可证明,再无任何打印机 Σ_i,使 $240<\Sigma_i(5)<501$。下面列出优胜者所构造的 Σ_1:见图2-1。

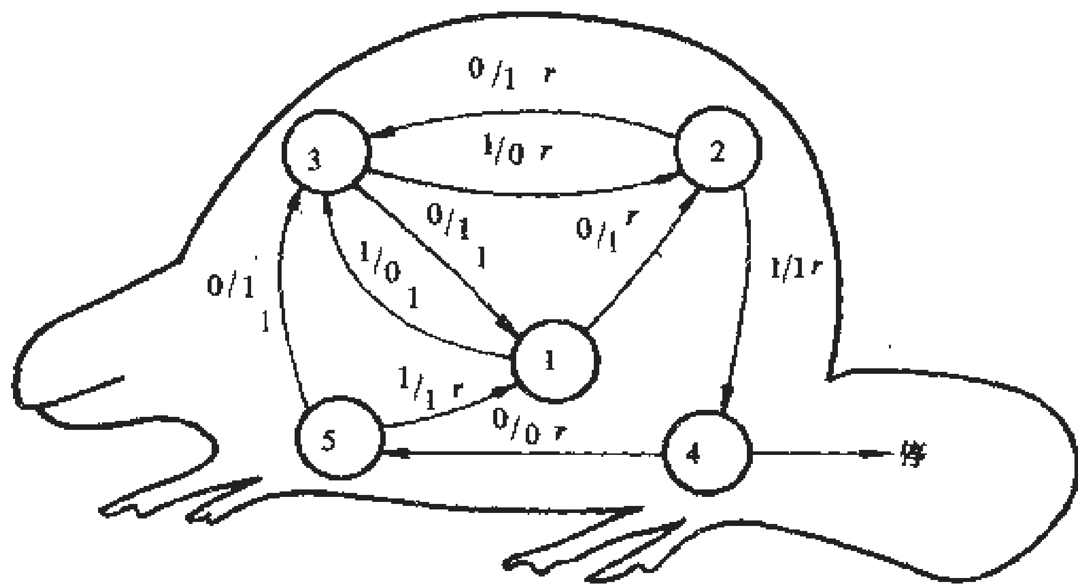


图2-1

有兴趣者请参阅《Scientific American, Feb. 1984》(中译本《科学》,1984年第7期)。

另外,《Handbook of Math. Logic》上一文"Elements of Recursion Theory"介绍说:$\Sigma(6) \geqslant 35$,$\Sigma(7) \geqslant 22961$,$\Sigma(8) \geqslant 8 \times 10^{44}$。

本书写作中，我们又见到 Buntrock 和 Marxen 构造出的 $\Sigma_3(5)=4096$。因而，$\Sigma(5)\geqslant 4096$。该 $\Sigma_3(5)$ 是移动了11798826步之后，才在带子上留下4096个“1”。他二人还构造了类似 Σ 的五个状态的打印机 $S_1(5)$，它移动步数特别多($S_1(5)=23554768$)。这 $S_1(5)$ 和上述 $\Sigma_3(5)$ 如下图。我们的学生王超编程序验证了 $\Sigma_1(5)$，$\Sigma_3(5)$ 和 $S_1(5)$。

$\Sigma_3(5)$

	0	1
q_1	$1rq_1$	$1rq_1$
q_2	$1lq_3$	$1lq_2$
q_3	$1rq_1$	$1lq_4$
q_4	$1rq_1$	$1lq_5$
q_5	$1lq_f$	$0lq_3$

$S_1(5)$

	0	1
q_1	$1rq_2$	$0lq_4$
q_2	$1lq_3$	$1rq_4$
q_3	$1lq_1$	$1lq_3$
q_4	$1rq_f$	$1rq_5$
q_5	$1rq_1$	$0rq_2$

我们将 n 个状态的移位次数最多的打印机记为 $S(n)$。Julstrom 于其1992年的文章中证得

$$S(n)\leqslant\Sigma(20n);$$

王克文和徐书润于1995年的文章中，将该结果改进为 $S(n)\leqslant\Sigma(10n)$；而杨瑞光、丁龙云和徐书润于其1997年的文章中又将该结果改进为 $S(n)\leqslant\Sigma(3n+c)$。Ben-Amram，Julstrom 和 Zwick 等人于1996年独立得到 $S(n)\leqslant(2n-1)\Sigma(3n+3)$。请参阅[30]，[31]，[61]，[62]和[63]。

正如王浩在其1981年的书中所说：“忙海狸问题是组合练习的一个取之不尽的源泉”。

在结束本小节之前，再介绍一种类似的定义非递归函数的方法。

称一种 Turing 机为**输入打印机**，其功能同打印机，只是其输入不是空白带，而是与打印机状态数相等的“1”的序列。

称一种 Turing 机为**相对于$\{f_0,f_1,\cdots,f_l\}$的扩充打印机**，其功能恰如输入打印机，只是在其任何状态均可调用计算函数 f_0，$f_1,\cdots,f_l$ 的 Turing 机

那么，令

$$\Phi^0(x)=1+\max l\{T^{(x+1)}(x+1)=l \mid T^{(x+1)}\text{ 为有 } x+1\text{ 个状态的输入打印机}\},$$

对 $k>0$，令

$$\Phi^k(x)=1+\max l\{T_1^{(x+1)}(x+1)=l \mid T_1^{(x+1)}\text{ 为有 } x+1\text{ 个状态的相对于}[\Phi^0(x),\cdots,\Phi^{(k-1)}(x)]\text{的扩充打印机}\},$$

则不难证明，$\Phi^k(x)$均非递归函数。读者可试证明之。详见"A Sequence of Uncomputable Functions"，by C. B. Dunham，in《ACM SIGACT》News，16:3(1984)。

3.6 Turing 机的子类——有穷自动机

当限定 Turing 机的移动符号集 M 中只有一个"l"时，则称这种 Turing 机为**有穷自动机**。而且规定：

(1)当给定一个输入符号序列 $\alpha\in A^*$，其读写头开始时总注视着 α 的最右一个符号；

(2)有穷自动机停机当且仅当其读写头移出字 α 时，并称此时带子上的内容即为有穷自动机的输出。

现在来研究有穷自动机的功能。

为确定和简单起见，不失一般性，假设

$$A=\{0,1,2\},\text{"0" 表示"空白"},$$
$$Q=\{q_1,\cdots,q_t\},$$

那么，对任输入符号序列 $i_t\cdots i_1 i_0(i_j\in\{1,2\},j=0,1,\cdots,t)$，配以非0的自然数 $X=\sum_{j=0}^{t} i_j\cdot 2^j$。称这种配数为**2缺0进制配数**。经这样的配数后，任一有穷自动机就可看成是自然数到自然数上的函数。

对有穷自动机的每条指令 $I_{i_1}(i_1=1,\cdots,m)$

$$(q_i,a_j,a_k,l,q_p),$$

可分别作二个自然数谓词

$$\mathscr{D}_{i_1}(y_1,x_1,x_2)\Longleftrightarrow x_1=i\wedge y_1=j\wedge x_2=p$$

和

$$\mathscr{B}_{i_1}(y_1,x_1,y_2)\Longleftrightarrow x_1=i\wedge y_1=j\wedge y_2=k,$$

它们分别表示有穷自动机的状态转换和输出。

将这有穷个 $\mathscr{D}_1,\cdots,\mathscr{D}_m$ 和 $\mathscr{B}_1,\cdots,\mathscr{B}_m$ 分别以"$\vee$"连接起来，即得

$$\mathscr{D}(y_1,x_1,x_2)\Longleftrightarrow\bigvee_{i_1=1}^{m}\mathscr{D}_{i_1}(y_1,x_1,x_2),$$

和

$$\mathscr{B}(y_1,x_1,y_2)\Longleftrightarrow\bigvee_{i_1=1}^{m}\mathscr{B}_{i_1}(y_1,x_1,y_2),$$

则它们完全表示了这有穷自动机的一步状态转换和一步输出。

易知，$\mathscr{D}$ 和 $\mathscr{B}$ 的特征函数均属于 Grzegorczyk 函数类 ε^0。

设 $\log^{(m)}(X)=\begin{cases}0, & \text{如 } X=0\\ X \text{ 的 } m \text{ 缺0进制表示式之位数}, & \text{如 } X\neq 0;\end{cases}$

$$K^{(m)}(X,t)=\begin{cases}X \text{ 的 } m \text{ 缺0进制表示式的末第 } t+1 \text{ 位数字}, & \text{如果 } X\neq 0;\\ 0, & \text{否则};\end{cases}$$

那么，相应于上述 $\mathscr{D}$ 和 $\mathscr{B}$，令

$$\begin{cases}G(X,0)=1\\ G(X,t+1)=\mu X_2\leqslant l\ \mathscr{D}(G(X,t),K^{(2)}(X,t),X_2),\end{cases}$$

$$F(X,t)=\mu y_2\leqslant 2\ \mathscr{B}(G(X,t),K^{(2)}(X,t),y_2),$$

则不难知道：如果 X 为有穷自动机的输入符号序列的配数时，则 $G(X,t)$、$F(X,t)$ 分别表示该有穷自动机在 t 步时的状态下标、输出符号。而

$$\Psi(X)=\sum_{i=0}^{\log^{(2)}(X)\dot{-}1}F(X,i)\cdot 2^i$$

即为这有穷自动机所对应的函数。那么有：

定理15 $\Psi(X)\in\varepsilon^1$。

为证明该定理，必须顺序证明下述函数或由受囿 μ-算子定义的函数，分别属于 ε^0 和 ε^1。这里，在列出的同时，仅给出几个例示，其他则请有兴趣的读者补证之，或请参考文献[15]。

*1—4 $X \dot{-} y, [X/y], \mathrm{rem}(X,y), \mathrm{s_g}(X) \in \varepsilon^0$(已有)；

*5—6 $X + \mathrm{s_g}(y), X \cdot \mathrm{s_g}(y) \in \varepsilon^0$；

*7若 $\varphi(X_1, \cdots, X_n, y) \in \varepsilon^0$，则 $\varphi^*(X_1, \cdots, X_n, X)$

$$= \mu y_{\leqslant X}[\varphi(X_1, \cdots, X_n, y) = 0] \in \varepsilon^0;$$

#1 $\mathrm{s_{g_k}}(X) = \begin{cases} 1, X = k \\ 0, 否则 \end{cases} \quad k = 1, \cdots, m$；

$\because \mathrm{s_{g_k}}(X) = \mathrm{s_g}((X+1) \dot{-} k) \dot{-} \mathrm{s_g}(X \dot{-} k)$，$\therefore \mathrm{s_{g_k}}(X) \in \varepsilon^0$；

#2 $X \cdot \mathrm{s_{g_k}}(y) \in \varepsilon^0, k = 1, \cdots, m$；

#3 $f_1(X, y, z) = z \dot{-} y \cdot z$

$\because$

$$\begin{cases} f_1(X, 0, z) = X \\ f_1(X, y+1, z) = f_1(X, y, z) \dot{-} z, \end{cases}$$ $\therefore f_1(X, y, z) \in \varepsilon^0$；

#4 $f_2(X, y, z) = X \dot{-} z^y \in \varepsilon^0$；

#5 $f_3(X, y, z) = X \dot{-} \sum_{i=0}^{y} z^i \in \varepsilon^0$；

#6 $\mathrm{long}^{(m)}(X) \quad \in \varepsilon^0$；

#7 $\log^{(m)}(X) = \mathrm{long}^{(m)}(X) \dot{-} 1 \quad \in \varepsilon^0$；

#8 $g^{(m)}(X) = m^{\log^{(m)}(X)} \in \varepsilon^0$；

#9 $H^{(m)}(X, t) = \begin{cases} 0, 如果\ X = 0 或\ t > \log^{(m)}(X) \\ 1, 否则； \end{cases}$

$\because H^{(m)}(X, t) = \mathrm{s_g}((\log^{(m)}(X) + 1) \dot{-} t) \cdot \mathrm{s_g}(X)$，

$\therefore H^{(m)}(X, t) \in \varepsilon^0$；

#10 $C^{(m)}(X, t)$ = 截去 X 的 m 缺0进制表示式的最低 t 位所余下的数

$\because \begin{cases} C^{(m)}(X, 0) = X \\ C^{(m)}(X, t+1) = [C^{(m)}(X, t) \dot{-} 1/m], \end{cases}$

$\therefore C^{(m)}(X, t) \in \varepsilon^0$；

#11 $\because K^{(m)}(X, t) = (\mathrm{rem}(C^{(m)}(X, t) \dot{-} 1, m) + 1) \cdot$

$$\mathrm{s_g}(H^{(m)}(X, t)),$$

$\therefore K^{(m)}(X, t) \in \varepsilon^0$；

因而，$G(X,t),F(X,t)\in\varepsilon^0$；

#12 $\varphi^{(m)}(X,y)=\sum_{i=1}^{m}i\cdot X\cdot s_{g_i}(y)\in\varepsilon^1$；

#13 若 $F(X,t)\in\varepsilon^0$，且 $F(X,t)\leqslant m$，则

$$\Psi_*^{(m)}(X,z)=\sum_{i=0}^{\log^{(m)}(z)}F(X,i)\cdot m^i\in\varepsilon^1;$$

这是因为：

$$\begin{cases}\Psi_*^{(m)}(X,0)=F(X,0)\\ \Psi_*^{(m)}(X,z+1)=\Psi_*^{(m)}(X,z)+\varphi^{(m)}(g_i^{(m)}(z+1),\\ \qquad\qquad F(X,\log^{(m)}(z+1)))\\ \qquad\qquad \cdot s_g(\log^{(m)}(z+1)\dot{-}\log^{(m)}(z)),\\ \Psi_*^{(m)}(X,z)\leqslant mz+m;\end{cases}$$

故 $\Psi^{(m)}(X)=\Psi_*^{(m)}(X,X)\in\varepsilon^1$；

然而，$\Psi^{(m)}(X)$即 $\sum_{i=0}^{\log^{(m)}(X)}F(X,i)\cdot m^i$，

所以 $\Psi(X)=\sum_{i=0}^{\log^{(2)}(X)}F(X,i)\cdot 2^i\quad\in\varepsilon^1$。

因为乘法不在 ε^1 中，所以有穷自动机所对应的函数没有乘法。可见其功能之弱。然而，至今人类所造出的计算机，实际上不过是某个有穷自动机而已。如何严格地论证这一点，请读者思考之。

3.7 计算复杂性与 $P\overset{?}{=}NP$ 问题

在1.3和1.6对递归函数的分层研究，就是一种计算复杂性研究；Ritchie 在他的博士论文中，对 G-函数分层的 ε^3（常称作初等函数类）又作了进一步的分层研究。具体言之，即以有穷自动机所相关的函数集作为分层的最小函数集，记为 $\mathscr{F}^0$，而后，以 $\mathscr{F}^0$ 中函数作为 Turing 机计算函数的空间界限（即所用的最大格子数），这样所计算的函数构成 $\mathscr{F}^1$，以此类推，结果得到：$\varepsilon^3=\bigcup_{i=0}^{\infty}\mathscr{F}^i$，$\mathscr{F}^i\subsetneqq\mathscr{F}^{i+1}(i=0,1,\cdots)$。类似地，可以时间（或步数）界限作为尺度，得

到原始递归函数集 $\mathfrak{R}$ 的另一分层，即 $\mathfrak{R}=\bigcup_{i=0}^{\infty}\mathfrak{G}^i,\mathfrak{G}^i\subsetneqq\mathfrak{G}^{i+1}(i=0,1,2,\cdots)$，而且对 $n\geqslant 3$，$\mathfrak{G}^n=\varepsilon^n$；因而可以说，这种分层可称作 G-函数集的时限（或步限）计算。详细内容请参阅[58]和[17]。

在上述的计算复杂性研究中，G-函数分层仅涉及函数值的大小，而并未和"计算"连起来，而 Ritchie 分层和作者所作的 $\mathfrak{G}$-分层研究，又是牵涉到两个计算模型的（或说两种理论计算机的）。下面将给出多种复杂性定义，并介绍有关结果。

设 T 为一给定的 Turing 机，且假设对其输入符号字已进行自然数编码，则 T 就可看成是自然数的函数。令

$$t_T'(x)=\begin{cases}\text{该 } T \text{ 对输入 } x \text{ 所作的步数，如果 } T \text{ 对 } x \text{ 停机}\\ \text{无定义，} \qquad \text{否则，}\end{cases}$$

$$S_T'(x)=\begin{cases}\text{该 } T \text{ 对输入 } x \text{ 所用的格子数，如果 } T \text{ 对 } x \text{ 停机}\\ \text{无定义，} \qquad \text{否则，}\end{cases}$$

则 t_T'，S_T' 分别称作 T 的时间和空间复杂度。

Blum 在1967年定义了所谓的抽象复杂度。如下：设 $\varphi_0,\varphi_1,\varphi_2,\cdots$ 是所有递归函数的枚举，它们的计算复杂度是指递归函数的序列

$$\Phi_0,\Phi_1,\Phi_2,\cdots,$$

且满足：(1) $\varphi_i(x)$ 有定义当且仅当 $\Phi_i(x)$ 有定义，$i=0,1,2,\cdots$；

(2) 若 $K(i,x,y)=\begin{cases}1,\text{当 } \Phi_i(x)=y \text{ 时，}\\ 0,\text{否则，}\end{cases}$

则 $K(i,x,y)$ 是一个完全定义的递归函数（即一般递归函数）。

那么，不难知道，上述定义的 Turing 机时间和空间复杂度是符合 Blum 的抽象复杂度定义的。

从现在至本节末，仅对**全停机的 Turing 机**进行研究。

对任给 Turing 机 T，令

$$t_T(n)=\max t_T'(x)(x \text{ 的长度 } |x|\leqslant n),$$

$$S_T(n)=\max S_T'(x)(|x|\leqslant n),$$

则 t_T, S_T 是另一种时间、空间复杂度；该种复杂度仅和输入的长度有关。

如下类似地定义所谓并行复杂度。

对任给 Turing 机 T，对任给输入 x，设 $r_T'(x)$ 为 T 对输入 x 计算时读写头移动方向改变的次数，称作循回次数。令

$$r_T(n) = \max r_T'(x)(|x| \leqslant n),$$

则 r_T 称作该 Turing 机 T 的并行复杂度。

易证下述两个定理。

定理16 对任给 Turing 机 T，有常数 C，使得：

$$t_T(n) = Cnr_T(n) \cdot S_T(n)。$$

定理17（线加速定理） 对任给 Turing 机 T，对任给常数 $C>0$，均可构造等价的 Turing 机 T', T''，使满足：

$$t_T(n) = Ct_{T'}(n), S_T(n) = CS_{T''}(n)。$$

该**定理16**叙述了时间、空间和并行复杂度之间的关系。粗略地说，时间复杂度是其他二者的积。

而**定理17**即所谓的线加速定理，其意思是说，就时、空而言，计算复杂度是可以“任意地”小或大的。更有下述所谓的“任意”加速定理。

定理18 设 $r(x)$ 是任给的全定义的递归函数，均有 Turing 机 T，使可构造另一等价的 Turing 机 T'，使得：

$$t_T(n) = r(t_{T'}(n))。$$

证明 略。

上述定义了几种计算复杂性测度，是否可信呢?为回答该问题，在本世纪70年代提出所谓的“大致等价性”（rouphly equivalent）和“相似性原理”(similarity principles)，其是说：对任二种计算模型，用其中之一表述的计算 A，可找到用另种模型表述的计算 B，且 B 可模拟 A，并在时间、空间、并行复杂度间，是多项式相关联的。该相似性或等价性的意义在于，若容许多项式相差的前提，则复杂度定义与计算模型无关。详请参阅文献 [24]。

“任意”加速定理（定理18）的涵义，是说任给的算法，都

无最好的程序。这就是说，一个函数的复杂性多种多样。初看起来，这似乎不太合理，因为这种复杂性的测度无标准可言了。上述这些话，在多种讲述加速定理的书中，都会有类似的说法。那么，到底这种复杂性定义还成立与否呢？

在文献［16］中，作者引入一种算子，由它出发，可对任意的递归函数集进行分层，且使得在较高层（或较大层）的函数，无论如何加速，也不会加速到较低层来。或者换句话说，这种分层才真正定义了函数的复杂性，即一函数的复杂性是由其所在的层刻画的，而不是靠一个函数刻划的。美国 MIT 的名教授 Meyer 和 Winklmann 在他们的文献［60］中，称一函数的复杂性能由一个“honest”函数序列来刻划，尽管他们并未对函数集进行分层。

在此之前介绍的 Turing 机，都是从计算函数的角度研究的。当然，包括计算特征函数在内。本节的余下部分我们讨论 Turing 机的可接受性，从而引出至今未曾解决的重要问题：$P=NP$？

设任给符号的有穷集 A，称 $L\subseteq A^*$ 为 A 上的语言。

例如，设 A 为 PASCAL 语言的所有符号构成之集（自然有穷），则所有的 PASCAL 程序即是 A 上的语言。再如，英文字母加上数字及若干标点之类的符号为符号集 A，则任一篇英文文章即构成一个 A 上的语言。

现在，对3.1有关 Turing 机的定义，再加以明确区分，有确定的 Turing 机，也有不确定的 Turing 机。事实上，那里在程序运行的定义中，规定 Turing 机只有一个初始状态，还规定程序的任二条指令，其前二元素不会完全相符（即在任何时刻，不会同时有两条或两条以上的指令可以执行）。我们称，经上述限制的 Turing 机为**确定**的 Turing 机。而当没有上述限制，即可以有一个初始状态集 $Q_1\subseteq Q$，且也可有若干指令，其前二元素完全相符，则称这种 Turing 机为**不确定的**。即它们可从任一个初始状态 $q_1\in Q_1$ 开始运行，且当有若干条指令可以执行时，就能选取其中的任一条指令执行。

为研究 Turing 机的可接受性，再从其状态集中划定一个接受

状态集$Q_a \subseteq Q$，且当Turing机转入此种状态时，也即刻停机。

设给定符号的有穷集A和以A作为符号集的Turing机T（其为确定的或不确定的，且有接受状态集$Q_a \subseteq Q$）。现在研究A上语言L的可接收性。

称$\alpha \in A^*$为T所接收，当且仅当T以α作为其带子上开始时的内容，使T可以运行且能转入接受状态$q \in Q_a$。

当然，对确定的T言，这个运行过程是确定的，最后转入停机状态或接受状态；而当T非确定时，无论从哪个初始状态出发，也无论运行中选取哪条可执行指令执行，只要最后能转入接受状态，即认为带上的开始时的内容是可接收的，否则认为是不可接收的。

称A上的语言L为T所接收，当且仅当

$$L = \{\alpha | \alpha \in A^*, \text{且 } \alpha \text{ 为 } T \text{ 所接收}\}。$$

因为任一确定的Turing机也可看成是不确定的，所以，确定的Turing机所能接收的语言必也为非确定的Turing机所接收。反过来，就任意的不加别的限制的Turing机而言，凡能被不确定的Turing机所接收的语言也能为确定的Turing机所接收。

现在，对Turing机的接受性加以限制。**称$\alpha \in A^*$能被Turing机T在多项式时间内接收**，当且仅当有多项式$p(x)$，而T接收α所用的指令次数$\leqslant P(\alpha)$。

称A上的语言L能被T在多项式时间内接收，当且仅当

$$L = \{\alpha | \alpha \in A^*, \text{且 } \alpha \text{ 为 } T \text{ 在多项式时间内接收}\}。$$

令$P = \{L | L \subseteq A^*$，且$L$能被确定的Turing机在多项式时间内接收$\}$，

$NP = \{L | L \subseteq A^*$，且$L$能被不确定的Turing机在多项式时间内接收$\}$，

那么，显然有，$P \subseteq NP$；但是否也有$P \supseteq NP$呢？这便成了长期来没有能够解决的大难题了。

对计算机科学理论中这一深刻的重大问题，不再作更进一步的介绍，有兴趣者可参阅有关文章，如：(1) The NP－Complete-

ness Column: An Ongoing Guide, by D. S. Johnson, J. of Algorithms, **Vol.** 11. **No.** 1, 144—151, 1990; (2) The structural Complexity Column, by J. Hartmanis, Bull. European Assoc. Theor. Comput. Sci（各集)。

3.8 Post 系统和 Turing 机

E. L. Post 为研究数学的推理系统,早在本世纪20年代初就提出并研究了几个所谓的 Post 系统,但由于种种原因,他的文章一直到40年代才得以发表。这里仅简单介绍一下他的规范系统(canonical system)。并分析该系统和 Turing 机的关系。

Post 规范系统是一个三元组(A,W_0,R),其中,A 是符号有穷集,$W_0 \subseteq A^*$,称作初始字集,而 R 是推导规则的有穷集,任 $r \in R$,形如

$$\xi_1 X_1 \xi_2 X_2 \cdots \xi_n X_n \xi_{n+1} \to \eta_1 X_{i_1} \eta_2 X_{i_2} \cdots \eta_m X_{i_m} \eta_{m+1},$$

且 $\xi_i, \eta_j \in A^*$ $(i=1,\cdots,n+1; j=1,\cdots,m+1)$,$X_k (k=1,\cdots,n)$称作变元,$X_{i_j} \in \{X_1,\cdots,X_n\} (j=1,\cdots,m)$。

现在介绍规范系统是如何运行的。

设任给 $\alpha \in W_0$,则将 α 按任推导规则中"→"的左边部分(以后简称**左部**)分解,若能为

$$\alpha = \xi_1 \delta_1 \xi_2 \delta_2 \cdots \xi_n \delta_n \xi_{n+1},$$

则称这分解与该推导规则左部相匹配,那么,即可使用该规则,从 α 得到

$$\beta = \eta_1 \delta_{i_1} \eta_2 \delta_{i_2} \cdots \eta_m \delta_{i_m} \eta_{m+1},$$

其中,δ_{i_j}是相应于 X_{i_j}从$\{\delta_1,\cdots,\delta_n\}$替换而来。此时记作 $\alpha \Rightarrow \beta$。对 β 又可再次使用推导规则进行推导。一直到不能再施行任何规则时止,所得到的字 γ,即称 γ 为 α 的推导结果。记作 $\alpha \overset{*}{\Rightarrow} \gamma$。令

$$W = \{\gamma \mid \alpha \overset{*}{\Rightarrow} \gamma, \alpha \in W_0\},$$

则W为该规范系统的导出集。

那么,我们有:

定理19　Turing 机所接受的语言类与规范系统导出集的类相同。

该定理的详细证明略去。这里仅作如下解释。

首先，若将 Turing 机的动态信息写成

$$u\ q_i\ a_j\ v,$$

并将相应的指令写成规范系统的推导规则，则易知：任一 Turing 机不过是一个规范系统。

其次，可以类似证明 Turing 机计算函数的递归性，证明任一规范系统的导出集的特征函数也是递归的。

这样，从上述研究可以看到，尽管递归函数、Turing 机和 Post 规范系统，从形式上看很不相同，但它们的计算、接受、推导能力却是相同的。在30年代末期，就有了这些系统（只是 Post 系统尚未公开发表，但还有 A. Church 的 λ-演算等），且证了它们的等价性。这便引起所谓的 **Turing－Church 论题**：任何一个可计算函数均为递归函数，或者说，任何一个可计算函数均为 Turing 机可计算。这个论题是无法给出数学上的严格证明的。因为“可计算函数”这不是一个数学概念。但是，当承认了这个论题，就可以说，“可计算”就是 Turing 机可计算，或者是递归可计算。下一节，我们将从分析 Turing 机定义的特殊性出发，给出另一种理想计算机 MMCM，而且从某种意义上说，这是最一般的理想计算机，也可以说是“可计算”的一个数学定义。

第四节　计算机的数学模型（MMCM）

上一节介绍了 Turing 机和 Post 系统，并证明了从接受性和导出性的意义讲，二者是相等价的（**定理19**）。本节，先从分析 Turing 机的特殊性和 Post 系统的非确定性出发，引出一般的理想计算机，称作计算机的数学模型，记为 MMCM。进而证 MMCM 和其他计算模型的等价性，并指出 MMCM 的一般性。以 MMCM 为背景，研究理想计算机的一些性质，并指出有穷性原则就是通过

有穷映射定义递归集和递归可枚举集的原则。

4.1 进一步分析 Turing 机和 Post 系统

在提示证明定理19时，曾建议将 Turing 机的动态信息写为 uq_ia_jv，这就是 Post 规范系统推导规则的左部形式。若有 Turing 机指令

$$(q_i, a_j, a_k, l, q_p),$$

则有推导规则：

$$ua_sq_ia_jv \longrightarrow uq_pa_sa_kv,$$

其中，a_s 为给定 Turing 机的任何符号。那么可知，将 Turing 机的各指令均类似地对应于若干条推理规则后，给定的 Turing 机一步步地运行，所作的各种动作，则这样构造的规范系统也能一步步地进行推理模拟。且这种规范系统的推理规则形式不外是

$$u\alpha v \longrightarrow u\beta v,$$

其中，u，v 是变元，α，β 是常数；这远比一般的推理规则形式

$$\xi_1x_1\xi_2x_2\cdots\xi_nx_n\xi_{n+1} \longrightarrow \eta_1x_{i_1}\eta_2x_{i_2}\cdots\eta_mx_{i_m}\eta_{m+1},$$

要简单和特殊得多。

实际上，可把 Turing 机的指令，看成每次仅对一个符号进行操作，把它改写成另外所需要的符号。比如，上述列出的指令，仅将 a_j 变成 a_k，而后即左移去注视和操作紧邻的符号了。

现在来看看 Post 规范系统。

首先，对运算对象所进行的操作，正像上边模拟 Turing 机一样，Post 规范系统可完成 Turing 机的功能，另外，它还可以同时作更多的操作，计有：

(1) ξ_1，ξ_2，…，ξ_{n+1}这是操作常数，将这些数同时变成常数 η_1，$\cdots\eta_m$，η_{m+1}；

(2) 将 X_1，…，X_n 诸变元所代入的数进行删除、考贝，并与 (1) 中得到的各常数 η_j ($j=1$，…，$m+1$) 进行一定次序的间隔排列。这就是该推理规则所得的最后结果。

因而，可以这样看：如果说 Turing 机仅有一个读写头，每次

仅对注视符号进行操作的话，那么，Post 规范系统即可看成有多个（实际是有限个）读写头，且对所有的注视符号进行操作，然后将产生的中间结果（即那些 η_i），和原来没有注视的部分，经过删除和考贝后，再进行重新排列，得到最后结果。而 Turing 机对原来没有注视的部分，既不删除、考贝，也不重新排列。

其次，再来分析一下 Turing 机作为计算模型时的确定性，而 Post 系统却不具有这种确定性。当然，Post 规范系统本来就不是作为计算系统，而是作为证明系统来设计的，故不确定性正是此种功能的要求。现在，我们则是一方面要利用规范系统对运算对象操作的一般性（以后将要论述这是最一般的），而又要排除它的不确定性，从而得到一个一般的计算系统。

Turing 机的确定性是通过要转去的状态去选择下一条要执行的指令。另方面，当把注视格中符号改写后，即左移一格或右移一格，而去寻找下一个操作符号（当然，选择指令时，除取决于状态，也取决于该新的操作符号）。而规范系统则不仅没有要转去的状态指示，也无明确指出如何去选择下一条推理规则；仅仅告诉，将所产生的最后结果去与另外的推导规则的左部相匹配，即去分解最后结果，确定哪些是操作符号（或操作字），哪些不是（而仅仅是要删除或要考贝并与将要得到的中间结果进行重新排序的部分）。如果仿照 Turing 机的读写头左移或右移一格，去找下条指令的操作符号，则规范系统可如下处理：(1)将所得最后结果中的若干段常数不去注视（相当于去掉若干读写头）；(2)增加注视最后结果中若干常数的左邻或右邻符号（相当于 Turing 机的左移或右移）；(3)将一常数分成二个常数（相当于增加读写头）。这样，对运算对象的处理就作好了。还可类似 Turing 机增加对状态的处理，则规范系统也就成了一类似 Turing 机的确定的计算系统。

经上述对 Post 规范系统的修改，得到了 MMCM.

4.2 MMCM 定义

先明确一些记号。

设 A 是任给定的符号集，则

$$\langle A\rangle_N = \{\alpha \mid \alpha \in A^*, \text{且 } |\alpha| \leqslant N\},$$

其中，N 为固定的自然数。

1)MMCM 的计算装置：与 Turing 机相类似，只是 MMCM 不只一个读写头，而是有不超过某一固定数的读写头，且有共同的状态记忆寄存器。

2)MMCM 的语言、程序：

设 $A^{(1)}$表示符号的有穷集，且对 $n \geqslant 2$,

$$A^{(n)} = (A^{(n-1)})^*,$$

且称 $A^{(n)}(n \geqslant 1)$为 n 级符号集，$A^{(n+1)}(n \geqslant 1)$也称作 n 级字集。

如 Turing 机一样，用 $A^{(n)}$和 $A^{(n+1)}$表示 n 级运算对象和结果。且对 n 级 MMCM，其带子的每个格子中都可存贮 $A^{(n)}$的一个符号。

现在，先讨论一级的 MMCM，如何对其计算装置操作。

设 $Q = \{q_1, \cdots, q_n\}$，称作状态的有穷集。其作用与 Turing 机相同。设 $V = \{v_1, \cdots, v_N\}$，称作变元符号集。

一级 MMCM 的程序是一映射：

$$P^{(1)}: E \times Q \longrightarrow E' \times Q,$$

其中，Q 为状态的有穷集，而 $E \subseteq \langle D\rangle_N$，$E' \subseteq \langle D'\rangle_N$，且

$$D = V \cup \overset{\downarrow}{A}{}^{(1)}, D' = V \cup \overset{\downarrow}{V} \cup A^{(1)} \cup \overset{\downarrow}{A}{}^{(1)},$$

这里，$\quad \overset{\downarrow}{A}{}^{(1)} = \{\overset{\downarrow}{a} \mid a \in A^{(1)}\}, \overset{\downarrow}{V} = \{{}^{\downarrow}v, v^{\downarrow}, {}^{\downarrow}v^{\downarrow} \mid v \in V\}$,

$P^{(1)}$还满足：对任给 $e \in E, q \in Q$，$P^{(1)}(e, q)$中的变元符号必在 e 中出现。

3)程序运行：类似 Turing 机，必须首先设定初始状态，比如说为 q_1，还要设定初始字，比如说对任给 $\alpha \in A^{(2)}$，必须假定 α 的哪些符号处在读写头注视之下。为了叙述方便，设有

$$P^{(1)}(v_1 \overset{\downarrow}{a}_2 v_2 \overset{\downarrow}{a}_5 v_3 \overset{\downarrow}{a}_9 v_4, q_1) = ({}^{\downarrow}v_2 b_1 b_2 v_4 b_3\ {}^{\downarrow}v^{\downarrow}_3 b_4 \overset{\downarrow}{b}_5 b_6, q_2), \tag{1}$$

并设开始时，给定字

$$\alpha = a_1\overset{\downarrow}{a}_2a_3a_4\overset{\downarrow}{a}_5a_6a_7a_8\overset{\downarrow}{a}_9a_{10}a_{11},$$

$\overset{\downarrow}{a}_2,\overset{\downarrow}{a}_5,\overset{\downarrow}{a}_9$表示有三个读写头，分别注视着它们，那么，经过所给的程序作用，得到字

$$\alpha' = a_3\overset{\downarrow}{a}_4b_1b_2a_{10}a_{11}b_3\overset{\downarrow}{a}_6a_7\overset{\downarrow}{a}_8b_4\overset{\downarrow}{b}_5b_6,$$

如果还有

$$P^{(1)}(v_1\overset{\downarrow}{a}_4v_2\overset{\downarrow}{a}_6v_3\overset{\downarrow}{a}_8v_4\overset{\downarrow}{b}_5v_5,q_2) = (\overset{\downarrow}{v}_3v_1,q_3), \qquad (2)$$

则又得到

$$\alpha'' = \overset{\downarrow}{a}_7a_3$$

……

依此类推，直到碰到最后状态为止。此时所得的结果字即称作该 MMCM 对所给开始字 α 的运行结果。并称，(1)中的 a_2,a_5,a_9为**操作符号**，而 b_1,b_2,b_3,b_4,b_5,b_6为**新生符号**，且显然有：新生符号是操作符号和状态 q_1的函数，当然，状态 q_2也是它们的函数。同样，(2)中的 a_4,a_6,a_8,b_5也是操作符号，而无新生符号，但状态 q_3是它们的函数，事实上，可将(1)和(2)分别改写为：

$$p_1(a_2,a_5,a_9,q_1) = b_1$$
$$p_2(a_2,a_5,a_9,q_1) = b_2$$
$$p_3(a_2,a_5,a_9,q_1) = b_3$$
$$\vdots$$
$$p_6(a_2,a_5,a_9,q_1) = b_6$$
$$p_7(a_2,a_5,a_9,q_1) = q_2$$

和

$$p_8(a_4,a_6,a_8,b_5,q_2) = q_3 \;,$$

也称这些 $p_1,p_2,\cdots,p_8$为 $P^{(1)}$的分解函数。

假设任意的 $i\leqslant n$ 级 MMCM 已经定义好了，现在来定义 $n+1$ 级 MMCM。

对 n 级 MMCM，若要对 $A^{(n)}$上字的若干元组进行运算时，可以给 $A^{(n)}$外增一个特殊符号“ * ”，得到 $A^{(n)}_*$，那么，即可用 $\alpha_1*\alpha_2*\cdots*\alpha_n$ 表示 $A^{(n)}$上字的 n 元组；自然，这时的 MMCM 已扩充为符

号集 $A_*^{(n)}$ 上了。

称 $A^{(n+1)}$ 上的函数是 n -级 MMCM 可计算的，如果有 n 级 MMCM，且对任给的该函数的自变量的值，作为该 n 级 MMCM 的开始时的字，运行它，最后碰到最后状态停机时，所得的结果字，恰为该函数的值。

那么，$n+1$级 MMCM 的定义，可类似 n 级 MMCM 的，只是其符号集当然为 $A^{(n+1)}$，且该 $n+1$级 MMCM 的程序 $P^{(n+1)}$的分解函数均为 n 级 MMCM 可计算函数。

另外规定，当运行 MMCM 时，对 $\overset{\downarrow\downarrow}{v}$ 代入，若 v 仅一个符号 a，则结果为 $\overset{\downarrow}{a}$；对 $\overset{\downarrow\downarrow}{v}$、$\overset{\downarrow}{v}$或 $\overset{\downarrow}{v}$代入，若 v 为“空字”，则结果为“空字”，相应的箭头也消失。

例 37 设 $A^{(1)}=\{0,1,\cdots,9,*\}$，$Q=\{q_1,q_2,q_3\}$，则如下的 MMCM μ 正好计算了普通的十进制加法函数。其程序为：(设 $i,j,k,h,h_1\in\{0,1,\cdots,9\}$)

(1) $(v_1\overset{\downarrow}{i}v_2\overset{\downarrow}{j},q_1)\longrightarrow(\overset{\downarrow}{v}_1\overset{\downarrow}{h}\overset{\downarrow}{v}_2k,q_2)$, $\quad i+j=hk$,

(2) $(\overset{\downarrow}{h}\overset{\downarrow}{*}v_1,q_2)\longrightarrow(hv_1,q_3)$, $\quad h\neq 0$,

(3) $(\overset{\downarrow}{o}\overset{\downarrow}{*}v_1,q_2)\longrightarrow(v_1,q_3)$,

(4) $(v_1\overset{\downarrow}{i}\overset{\downarrow}{j}\overset{\downarrow}{*}v_2,q_2)\longrightarrow(\overset{\downarrow}{v}_1\overset{\downarrow}{h}\overset{\downarrow}{*}kv_2,q_2)$, $\quad i+j=hk$,

(5) $(\overset{\downarrow}{i}v_1\overset{\downarrow}{j}v_2,q_2)\longrightarrow(\overset{\downarrow}{h}\overset{\downarrow}{v}_1kv_2,q_2)$, $\quad i+j=hk$,

(6) $(v_1\overset{\downarrow}{i}\overset{\downarrow}{j}v_2\overset{\downarrow}{h}_1v_3,q_2)\longrightarrow(\overset{\downarrow}{v}_1\overset{\downarrow}{h}\overset{\downarrow}{v}_2kv_3,q_2)$, $i+j+h_1=hk$,

对这 μ 言，$N=6$。

比如，要求125＋2508，则上述 μ 的运行过程为：

$$\underbrace{12}_{v_1}\overset{\downarrow}{5}\,\underbrace{*\,250}_{v_2}\overset{\downarrow}{8}\xRightarrow[q_2]{q_1,(1)}\underbrace{1}_{v_1}\overset{\downarrow}{2}\overset{\downarrow}{1}\,\underbrace{*\,25\overset{\downarrow}{0}3}_{v_2},\qquad 5+8=13,$$

$$\underbrace{1}_{v_1}\overset{\downarrow}{2}\overset{\downarrow}{1}\,\underbrace{*\,25}_{v_2}\underbrace{\overset{\downarrow}{0}3}_{v_3}\xRightarrow[q_2]{q_2,(6)}\underbrace{\overset{\downarrow}{1}\overset{\downarrow}{0}}_{v_1}\,\underbrace{*\,2\overset{\downarrow}{5}}_{v_2}\underbrace{33}_{v_3},\qquad 2+1+0=03,$$

$$\underbrace{\quad}_{v_1}\overset{\downarrow}{1}\overset{\downarrow}{0}\underbrace{*\,2}_{v_2}\overset{\downarrow}{5}\underbrace{33}_{v_3}\xRightarrow[q_2]{q_2,(6)}\underbrace{\quad}_{v_1}\overset{\downarrow}{0}\underbrace{*\,\overset{\downarrow}{2}}_{v_2}6\underbrace{33}_{v_3},\qquad 1+0+5=06,$$

$$\overset{\downarrow}{0}\underbrace{*}_{v_1}\underbrace{\overset{\downarrow}{2}633}_{v_2}\xRightarrow[q_2]{q_2,(5)}\overset{\downarrow}{0}\underbrace{\overset{\downarrow}{*}}_{v_1}\underbrace{2633}_{v_2},\qquad 0+2=02,$$

$$\overset{\downarrow}{0}\overset{\downarrow}{*}\underbrace{2633}_{v_1}\xRightarrow[q_3]{q_2,(3)}\underbrace{2633}_{v_1}。$$

停止运行。

例38 试编制计算乘法的MMCM。(练习)

4.3 MMCM 和其他理想计算机

因为 Turing 机的任给指令

$$(q_i,a_j,a_k,l,q_p),$$

可改写为

$$(v_1\overset{\downarrow}{a}_jv_2,q_i)\longrightarrow(\overset{\downarrow}{v}_1a_kv_2,q_p),$$

故明显地有:

定理20 对任给 Turing 机,均可构造等价的 MMCM。

现在介绍理想计算机 URIM:

(1)计算装置:由无界多个寄存器 $R_i(i=1,2,3,\cdots)$和指令计数器 K 组成;

(2)语言:$A=\{$自然数$\}$,每个自然数可存贮在一个寄存器 R_i 中;所以,R_i 是用来寄存开始时输入,中间计算结果和最后计算结果的。

URIM 的指令有如下几种:

Z_i:将 R_i 寄存器内容清为0;

S_i:将 R_i 寄存器内容加1;

$T(R_i,R_j)$:将 R_i 内容送入 R_j 中;

J_j:转到第 j 条指令执行;

$J_j[R_i]$:当 R_i 内容为0时,转第 j 条指令执行,否则,K 内容加1(即执行下条指令)。

(3)程序及运行：

URIM 程序 P 为一指令序列 $I_0, I_1, \cdots, I_{h-1}$，指令的下标表示指令的位置。

开始时，K 内容 $k=0$，R_i 中内容除给定输入 r_i 不为0外，其他均为0；因而，程序总是从第0条指令 I_0开始运行。

程序运行中未碰到转移指令时，即顺序执行(即 K 内容加1)，否则，转第 j 条指令(当运行 J_j 时)或判定 R_i 内容决定转不转第 j 条指令(当运行 $J_j[R_i]$时)，一旦转移时，K 的内容即改为 j，否则 K 内容加1。

当 K 中内容 $k \geqslant h$ 时，即告停机。

那么，不难证得下述定理：

定理21　对任给的 URIM 程序 P，均可构造2级 MMCM，使二者的功能相同。

4.4　MMCM 的一般性

从4.3已可看出，对 Turing 机，对 Post 规范系统（确定的），对 URIM，很易构造等价的 MMCM。事实上，就作者所知的计算模型而言，在某种意义上说，都是 MMCM 的特例。

本小节，我们来分析一下 MMCM 对运算对象的操作是最一般的。

无论什么计算，总要有计算对象，这些对象即构成计算域，且一般而言，这对象所构成的计算域是无穷的。一方面，随所给的计算对象 α 之不同，其长度是无界的，另方面，随所给 α 之不同，所进行的计算步数也是无界的。这两个无界正反映了计算时所用的空间和时间的无界性。这正说明计算的能力。

但是，计算的程序表述又必须有穷。因而，在从 α 到 α' 的一步计算中，必须要有常数 S，使在这一步计算中，至多有 α 的 S 个子部分发生改变，且改变部分的长度又必须有界。再注意到，在每步计算中，改变部分和不改变部分必然相间，故至多有 S 个长度无界的子部分不发生改变。在 MMCM 定义中操作符号和新生符号

正说明这种改变的部分，而变元正说明这种不改变的部分。读者也可能意识到这里术语的有趣性：在一步计算中要改变的是常符号，而不改变的却是变元所代表的。

读者可能要问，MMCM 的程序 $P^{(1)}$的定义中，定义域 $E\times Q$ 和值域 $E'\times Q$ 均有穷，为什么反会得到计算无穷对象域的MMCM呢?这里的关键是使用了“变元的**代入**”，而计算的复杂性又是由于步数的无界性所造成。在另外一些描述“产生”的计算模型中，尽管没有使用“变元代入”，但步数的无界性和不终止性则起了关键的作用。

4.5　递归集、递归可枚举集和通用 MMCM

就可计算理论（或理想计算机等）的总体的实质是对所谓的递归集、递归可枚举集以及不可判定的集合的存在性证明。这些结果是可在各种计算模型中描述和证明的。这里，主要是以MMCM为工具来进行。

称一集合 $B\subseteq A^{(2)}$为**递归可枚举集**，如果有一总会停机的MMCM μ，使

$$B=\{\alpha' \mid \text{有 } \alpha\in A^{(2)},\text{使 } \mu(\alpha)=\alpha'\},$$

这里，μ（α）表示以 α 作为开始字，μ 对它运行的结果。**空集也规定为递归可枚举集**（从 Post，1944）。

那么，有下述定理22。

定理22　显然有：

(1)$A^{(2)}$递归可枚举；

(2)$\{\alpha*\mu(\alpha)\mid\alpha\in A^{(2)},\mu$ 为总停机的 MMCM$\}$递归可枚举。

定理23　$A,B\subseteq A^{(2)}$，且 A,B 均递归可枚举，则 $A\cup B$ 也递归可枚举。

证明　由 A,B 递归可枚举，故有总停机的 MMCM μ_1,μ_2，使

$$A=\{\alpha' \mid \text{有 } \alpha\in A^{(2)},\text{使 } \mu_1(\alpha)=\alpha'\},$$

$$B=\{\alpha' \mid \text{有 } \alpha\in A^{(2)},\text{使 } \mu_2(\alpha)=\alpha'\}。$$

不妨设 $A^{(1)}=\{a_1,a_2,\cdots,a_m\}$。那么，如下构造一个 MMCM μ，其对

任一开始字α进行判别：

若$\alpha=a_i(i=1,\cdots,m)$，则μ转去执行μ_1，

若$\alpha=\beta a_1(\beta\neq$空字$\varepsilon)$，则将$\alpha$改为$\beta$，转去执行$\mu_1$；

若$\alpha=\beta a_2(\beta\neq\varepsilon)$，则将$\alpha$改为$\beta$，转去执行$\mu_2$；

若$\alpha=\beta a_i(\beta\neq\varepsilon,i\neq 1,2)$，则转去执行$\mu_2$。

易知，该μ即所求。证毕。

为证递归可枚举集对“$\cap$”运算也封闭，则没有**定理23**那么简单了。为达此目的，要作下述一些准备工作。

定理24 设A递归可枚举，则可构造总停机的μ，使$A=\{\alpha|\alpha\in A^{(2)}$，有$\alpha'\in A^{(2)}$，使得$\mu(\alpha*\alpha')=a_1\}$，其中，$a_1\in A^{(1)}$。反之，亦然。

证明 由A递归可枚举，则有μ_1，使

$$A=\{\alpha'\mid 有\ \alpha,\mu_1(\alpha)=\alpha'\},$$

那么，构造μ，使对任给的$\alpha*\alpha'$，均转μ_1'，运行结果为$\alpha*\alpha'*\mu_1(\alpha')$（这$\mu'_1$是从$\mu_1$构造的），此时，不再停机，而是接着比较“$\alpha$”和“$\mu_1(\alpha')$”，若相同，则结果为$a_1$，否则，结果为$a_2$；然后，即停机。易知，这$\mu$即所求。反之很明显。证毕。

定理25 设$A_1^{(1)}=\{a_1,\cdots,a_m\}$，$A_2^{(1)}=\{a_1\}$，而编码函数$\ulcorner\ \urcorner$：$A_1^{(2)}\longrightarrow A_2^{(2)}$；那么，$B_1\subseteq\{\alpha|\alpha\in A_1^{(2)}\}$递归可枚举当且仅当$B_2=\{\ulcorner\alpha\urcorner|\alpha\in B_1\}$递归可枚举。

证明 由B_1递归可枚举，故有 MMCM μ_{B_1}，使$B_1=\{\alpha|\alpha\in A_1^{(2)}$，有$\alpha'\in A_1^{(2)}$，$\mu_{B1}(\alpha')=\alpha\}$，

那么，构造μ_{B_2}：使对任$\alpha\in A_2^{(2)}$，先转逆编码 Turing 机，$\mu_{\ulcorner\urcorner}^{-1}$，得$\mu_{\ulcorner\urcorner}^{-1}(\alpha)$，而后转$\mu_{B_1}$，得到$\mu_{B_1}(\mu_{\ulcorner\urcorner}^{-1}(\alpha))$，再转编码 Turing 机$\mu_{\ulcorner\urcorner}$，得到

$$\mu_{\ulcorner\urcorner}(\mu_{B_1}(\mu_{\ulcorner\urcorner}^{-1}(\alpha))),$$

则

$$B_2=\{\alpha'|\alpha'\in A_2^{(2)},有\ \alpha\in A_2^{(2)},使\ \mu_{B_2}(\alpha)=\alpha'\};$$

反之，显然成立。证毕。

现在研究一下 Cantor 配对函数 J，其将自然数的序偶和自然数建立了1—1对应的关系。当然，实际上这也是一种编码函数。

比如，按下述次序与自然数建立对应：

第0组(0,0)

第1组(0,1),(1,0),(1,1)

第2组(0,2),(2,0),(1,2),(2,1),(2,2)

第3组(0,3),(3,0),(1,3),(3,1),(2,3),(3,2),(3,3)

⋮

即将它们按由上而下、由左至右的次序与自然数0,1,2,3,…一一相配，那么，这个配对函数 $J(x,y)$，即满足

$$J(0,0)=0,$$
$$J(0,1)=1, J(1,0)=2, J(1,1)=3,$$
$$J(0,2)=4, J(2,0)=5, J(1,2)=6, \cdots。$$

且与这 $J(x,y)$ 相对应的函数 K,L，

$$K(J(x,y))=x,$$
$$L(J(x,y))=y,$$

均是原始递归的。试练习写出它们的定义式。

这样，由**定理9**和**20**，可以构造相应于 K 和 L 的 MMCM μ_K 和 μ_L，使：

z	|	||	|||	||||	|||||	||||||	|||||||	…
$\mu_K(z)$	|	|	||	||	|	|||	||	…
$\mu_L(z)$	|	||	|	||	|||	|	|||	…。

设有 $A,B\subseteq A^{(2)}$，均递归可枚举，那么，由**定理24**，可分别构造总停机的 μ_A 和 μ_B，使

$$A=\{\alpha \mid 有\ \alpha' \in A^{(2)}, 使\ \mu_A(\alpha * \alpha')=a_1\},$$
$$B=\{\alpha \mid 有\ \alpha' \in A^{(2)}, 使\ \mu_B(\alpha * \alpha')=a_1\};$$

由 μ_A,μ_B 和 μ_K,μ_L，构造一个 MMCM，$\mu_{A\cap B}$，如下：

$\mu_{A\cap B}$对任给的 $\alpha * \beta$ 作为开始字动作；

若 $\beta = a_1 \cdots a_1$，则 $\mu_{A\cap B}$先执行 μ_K，结果为：

$$\alpha * \beta * \mu_K(\beta),$$

然后再转 μ_A，结果为：

$$\alpha * \beta * \alpha * \mu_K(\beta) * \mu_A(\alpha * \mu_K(\beta)),$$

接着又转 μ_L, μ_B，结果为：

$$\alpha * \beta * \alpha * \mu_K(\beta) * \mu_A(\alpha * \mu_K(\beta)) * \alpha * \mu_L(\beta) * \mu_B(\alpha * \mu_L(\beta))$$

判断 $\mu_A(\alpha * \mu_K(\beta)) = \mu_B(\alpha * \mu_L(\beta)) = a_1$，

若成立，即输出为 a_1，

否则，即输出为 a_2；

若 $\beta \neq a_1 \cdots a_1$，则 $\mu_{A\cap B}$输出为 a_2。

那么，

$$A \cap B = \{\alpha \mid 有\ \beta, 使\ \mu_{A\cap B}(\alpha * \beta) = a_1\} \qquad (*)$$

故 $A \cap B$ 也为递归可枚举集。

兹说明(*)的正确性。因为 $\beta = a_1 \cdots a_1$ 可表示任何自然数，故由 K 和 L，只要有 $\beta = a_1, \cdots, a_1$，即会有序偶$(\mu_K(\beta), \mu_L(\beta))$，所以，只要有 α_1，

使 $\mu_A(\alpha * \alpha_1) = a_1$(即 $\alpha \in A$)，

且有 α_2，使 $\mu_A(\alpha * \alpha_2) = a_1$(即 $\alpha \in B$)，

即会有 β，$\mu_K(\beta) = \alpha_1$，$\mu_L(\beta) = \alpha_2$，故此 β 即会使 $\mu_{A\cap B}(\alpha * \beta) = a_1$，即 $\alpha \in A \cap B$(请注意要使用**定理21**)。

由上述，即得：

定理26 递归可枚举集对"$\cap$"封闭。

现在讨论限制更多的集合，称作递归集。

称集合 $B \subseteq A^{(2)}$ 为**递归集**，如果有一总会停机的 MMCM μ，使得

$$B = \{\alpha \mid \mu(\alpha) = a_1, \alpha \in A^{(2)}, a_1 \in A^{(1)}, 且\ a_1\ 固定\}。$$

从定义可知：设给定递归集 B，则对任给的 $\alpha \in A^{(2)}$，可能行地判定是否有 $\alpha \in B$。因为这可通过运行 $\mu(\alpha)$，其在有穷步可停机，看其结果是否为 a_1，即可确定是否有 $\alpha \in B$。但对给定的递归可枚

举集 B，却不一定能判定是否有 $\alpha\in B$。因为当 B 不空时，尽管可以不断地运行，

$$\mu(\alpha_0),\mu(\alpha_1),\mu(\alpha_2),\cdots,\mu(\alpha_i),\cdots$$

其中，各 $\alpha_i\in A^{(2)}(i=0,1,\cdots)$。即使运行了相当多次，但在没有得到 α 时，也不可以就说 $\alpha\notin B$，因为说不定在以后的某个 α_j 时，有 $\mu(\alpha_j)=\alpha$。读者不难理解，即使按**定理24**所给的递归可枚举集的等价定义，也同样不能判定对任给的 α，是否有 $\alpha\in B$。

但有下述定理。

定理27 若 B 递归，则 B 递归可枚举。

证明 从递归集的定义，知空集 ϕ 递归，而从递归可枚举集定义，知 ϕ 也递归可枚举。故不妨设有 $\alpha_0\in B$，那么，由定义，

$$B=\{\alpha|\alpha\in A^{(2)},\mu(\alpha)=a_1,a_1\in A^{(1)},\text{且 } a_1\text{固定}\},$$

构造 μ'，使对任 $\alpha'\in A^{(2)}$，运行 $\mu(\alpha')$，那么，

$$\text{当 } \mu(\alpha')=a_1 \text{ 时，就让 } \mu' \text{ 输出 } \alpha',$$

$$\text{当 } \mu(\alpha')\neq a_1 \text{ 时，就让 } \mu' \text{ 输出 } \alpha_0;$$

显然有，$B=\{\alpha|\alpha\in A^{(2)},\text{有 } \beta\in A^{(2)},\mu'(\beta)=\alpha\}$，故 B 递归可枚举。证毕。

定理28 递归集对“$\cup$”、“$\cap$”、“补”运算都是封闭的。

证明 易证，略。

定理29 $B\subseteq A^{(2)}$，那么，B 递归当且仅当 B 和 $A^{(2)}-B$ 递归可枚举。

证明 B 递归，由**定理28** 关于递归集对补运算的封闭性，故 $A^{(2)}-B$ 也递归，再由**定理27**，故 B 和 $A^{(2)}-B$ 递归可枚举。

反过来，由 B 和 $A^{(2)}-B$ 递归可枚举，故有二个 MMCM μ_1和 μ_2，使

$$B=\{\alpha|\text{ 有 } \alpha',\mu_1(\alpha*\alpha')=a_1\},$$

$$A^{(2)}-B=\{\alpha|\text{ 有 } \alpha',\mu_2(\alpha*\alpha')=a_1\},$$

当然，其中的 $\alpha'\in A^{(2)}$；易知，可构造相应于 μ_1和 μ_2的 μ'_1和 μ'_2，使对任 $\alpha\in A^{(2)}$，有

$$\mu_i'(\alpha * \ulcorner\alpha'\urcorner) = \mu_i(\alpha * \alpha'),$$

其中 $\ulcorner\ \urcorner : A^{(2)} \to \{a_1\}^*$。

故上述 B 和 $A^{(2)}-B$ 可分别为

$$B = \{\alpha \mid \text{有 } \alpha' \in \{a_1\}^*, \mu_1'(\alpha * \alpha') = a_1\},$$

$$A^{(2)} - B = \{\alpha \mid \text{有 } \alpha' \in \{a_1\}^*, \mu_2'(\alpha * \alpha') = a_1\},$$

那么，由于 B 和 $A^{(2)}-B$ 互补，故对任给 $\alpha \in A^{(2)}$ 或者 $\alpha \in B$，或者 $\alpha \in A^{(2)}-B$，也就是说，总会有 $\alpha' \in \{a_1\}^*$，使得或者 $\mu_1'(\alpha * \alpha') = a_1$，或者 $\mu_2'(\alpha * \alpha') = a_1$。

所以，可如下构造 μ：

对任给 α，先计算 $\mu_1'(\alpha * \varepsilon)$，再计算 $\mu_2'(\alpha * \varepsilon)$，然后判定是否有一个为 a_1，若前者为 a_1，则输出 a_1，停机，若后者为 a_1，输出 a_2，停机；若二者均不为 a_1，则再计算

$\mu_1'(\alpha * a_1)$和 $\mu_2'(\alpha * a_1)$，

同上述一样去判定，去动作，若二者均不为 a_1，则再计算

$$\mu_1'(\alpha * a_1 a_1) \text{ 和 } \mu_2'(\alpha * a_1 a_1),$$

依此类推。

由于总会有 $\alpha' \in \{a_1\}^*$，使得或者

$$\mu_1'(\alpha * \alpha') = a_1 \quad \text{或者 } \mu_2'(\alpha * \alpha') = a_1,$$

故上述 μ 总会停机。那么，不难知道，有

$$B = \{\alpha \mid \mu(\alpha) = a_1, \quad \alpha \in A^{(2)}\},$$

因而，B 递归。证毕。

现在来研究通用 MMCM 与递归不可解性。

如3.4通用 Turing 机一节的研究，可建立下述定理。

定理30　可构造 $A^{(1)} \cup \{*\}$上的通用 MMCM，μ，其他为一个 MMCM，使对任给的 $A^{(1)}$上的 MMCM $\mu_i(i=1,2,3,\cdots)$，对任给的 $\alpha \in A^{(2)}$，满足

$$\mu(* \ulcorner\mu_i\urcorner * \alpha) = \mu_i(\alpha),$$

其中，$\ulcorner\ \urcorner$ 为一编码函数。

有兴趣的读者可去构造通用的 μ。我们所构造的 μ 单其指令

要有70条之多，较之通用 Turing 机复杂多了。

定理31 “对任给的 MMCM μ_i，对任给的字 $\alpha\in A^{(2)}$，是否有 $\mu_i(\alpha)$停机”，这是递归不可解的。即：没有一个全定义的（即总停机的）且仅取0或1为值的 MMCM μ'，使得，

$\mu'(*\ulcorner\mu_i\urcorner *\alpha)=1$当且仅当 $\mu_i(\alpha)$停机。

证明 由**定理26**，易构造 MMCM μ，使对任 MMCM μ_i，$\alpha\in A^{(2)}$，满足：

$$\mu(*\ulcorner\mu_i\urcorner *\alpha)=\mu_i(*\ulcorner\mu_i\urcorner *\alpha)。\tag{1}$$

现在用反证法来证该定理，且主要是对角线方法的运用。

设该定理中的问题可解（递归可解），即：有一个全定义的且仅取0或1为值的 MMCM μ，使得，

$\mu'(*\ulcorner\mu_i\urcorner *\alpha)=1$当且仅当 $\mu_i(\alpha)$停机，

（再由**定理30**）当且仅当 $\mu(*\ulcorner\mu_i\urcorner *\alpha)$停机。特别地，有：

$$\mu'(*\ulcorner\mu_i\urcorner *\alpha_i)=1 \text{ 当且仅当 } \mu(*\ulcorner\mu_i\urcorner *\alpha) \text{ 停机} \tag{2}$$

（这里，实际上已假定有二个序列：

$\mu_1,\mu_2,\mu_3,\cdots$（MMCM 的序列）

和

$\alpha_1,\alpha_2,\alpha_3,\cdots$（输入字的序列））

那么，由 μ' 为全停机且仅取0或1为值的 MMCM，则不难构造另一 MMCM μ''，使，

$$\mu''(*\ulcorner\mu_i\urcorner *\alpha_i)=\begin{cases}\text{不停机，若 } \mu'(*\ulcorner\mu_i\urcorner *\alpha_i)=1;\\ \text{停机，否则(即 } \mu'(\cdots)=0);\end{cases}\tag{3}$$

故在 MMCM 的序列中，必有一个 MMCM

$$\mu_K=\mu'',\tag{4}$$

因而，

$\mu(*\ulcorner\mu_k\urcorner *\alpha_k)$停机（由(2)）当且仅当 $\mu'(*\ulcorner\mu_k\urcorner *\alpha_k)=1$，

（由(3)）当且仅当 $\mu''(*\ulcorner\mu_k\urcorner *\alpha_k)$ 不停机，

（由(4)）当且仅当 $\mu_k(*\ulcorner\mu_k\urcorner *\alpha_k)$ 不停机，

但是，由(1)，又有：

$$\mu(* \ulcorner \mu_k \urcorner * \alpha_k)\text{停机，当且仅当}$$

$$\mu_k(* \ulcorner \mu_k \urcorner * \alpha_k)\text{停机，}$$

这与上述结果相矛盾。

故**定理31**证毕。

定理31的结果称作**停机问题的不可解性**，严格地说：停机问题的递归不可解性。意思即，不存在一个有穷描述的方法，使能判定对任给的理想计算机，对任给的它的输入，该理想计算机对该输入停止与否。

在**3.5**中证明了忙海狸 $\Sigma(x)$ 的非递归性，是通过它的值随 x 的增长，非常迅速地增长。类似证明 Ackermann 函数非原始递归性一样。但到底其与已有的递归不可解性问题有何关系呢？下面所引文章中证明了：Σ 的不可计算性等价于停机问题的递归不可解性。请参见：A Bound on the Shift Function in Terms of the Busy Beaver Function, by B, A, Julstrom, ACM SIGACT News, **Vol.** 23, **No.** 3, summer 1992, pp. 100—106.

4.6 递归可枚举集与递归不可解性

上一小节，递归可枚举集的二个等价定义都是从全停机的 MMCM 而来的。实际上，也可通过非全停机的 MMCM 来定义。

在**3.2**中，证明了任何 Turing 机所计算的函数均为递归函数，即

$$\varphi(x_1,\cdots,x_n) \simeq (\mu z[\theta(\tau_n(x_1,\cdots,x_n,1,1,1),(z)_0) = \tau_{n+1}(x_1,\cdots,x_n,(z)_1,0,(z)_2,(x)_3)])_1 .$$

当 $n=1$ 时[1]，即

$$\varphi(x) \simeq (\mu z[\,|\theta(\tau_1(x,1,1,1),(z)_0) - \tau_2(x,(z)_1,0,(z)_2,(z)_3)| = 0])_1,$$

且仅用了一次 μ-算子，其他均为原始递归算子或函数。那么，显然

1) 易将递归集、递归可枚举集推广到任意 n-序组。

有：

$\varphi(x)$有定义（亦即 Turing 机停机），当且仅当

有 z，使，

$$\overbrace{|\theta(\tau_1(x,1,1,1),(z)_0) - \tau_2(x,(z)_1,0,(z)_2,(z)_3)|}^{f(x,z)} = 0,$$

故令 $B=\{x|\varphi(x)$有定义$\}$，则 $B=\{x|$有 z，使 $f(x,z)=0$，f 全定义，f 递归$\}$。

结合递归函数、Turing 机、MMCM 的等价性，有：

定理32 任给 MMCM μ，有 μ'（全停机），使

$\{\alpha|\mu(\alpha)$有定义（即停机）$\}=\{\alpha|$有 α'，使 $\mu'(\alpha*\alpha')=a_1\}$，故由**定理24**，$B=\{\alpha|\mu(\alpha)$停机$\}$是递归可枚举集。

反过来也有：

定理33 任给递归可枚举集 B，均可构造 MMCM μ'，使 $B=\{\alpha|\mu'(\alpha)$停机$\}$。

证明　由**定理24**，可设 $B=\{\alpha|$有 α'，使 $\mu(\alpha*\alpha')=a_1\}$，$\mu$ 总停机。那么，可这样构造 μ'，使 $\mu'(\alpha)=\mu\alpha'[\mu(\alpha*\alpha')=a_1]$，结合**定理29**的证明，知这样的 μ' 是可以构造出的，只是这里的 μ' 不一定总会停机。证毕。

由**定理32、33**，又得到递归可枚举集的一个等价定义，即 B 递归可枚举当且仅当有 MMCM μ，使 $B=\{\alpha|\mu(\alpha)$停机$\}$（用递归函数的术语讲，即递归可枚举集与任一递归函数的定义域相等）。

通过 MMCM μ 的定义域，当然也构成一个递归可枚举集。那么，由**定理27**，立得：

定理34 有递归可枚举而非递归的集合。

现在简单介绍一下 Hilbert 第十问题。

1900年，Hilbert 在世界数学家大会上向廿世纪数学家挑战，提出的23个数学问题。第十问题是说：是否有一个给定的过程，使对任给的 Diophantine 方程（即整系数多项式方程 $P(x_1,\cdots x_n)=0$），在有限步运算后，判定其有无整数解。

经过无数数学家的辛勤工作，最后在1970年由苏联的年轻的

数学家(当时,他还是学生)Y. Matijasevič 给该问题以否定解答。即证明没有一个给定的满足这里要求的过程。对这个问题作出重要贡献的,还有 M. Davis,J. Robinson 和 H. Putnam。这里仅简单介绍一下解决的大体过程。详请参阅:"Hilbert's Tenth Problem, Diopahtine Equations:Positive Aspects of A Negative Solution", by M. Davis,Y. Matijasevič and J. Robinson, *in*《Math. Developments Arising From Hilbert Problems》,1976。(这是美国数学会1974年的特邀报告集)。

称自然数 n 元组集合 $S=\{(x_1,\cdots,x_n)\mid$ 有 $y_1,\cdots,y_m$,使 $[P(x_1,\cdots,x_n,y_1,\cdots,y_m)=0]\}$ 为 Diophantine 集(简称 D-集),其中,$P(x_1,\cdots,x_n,y_1,\cdots,y_m)$ 为多项式。

例如,$S_1=\{x\mid$ 有 y,z,使 $x=y(2z+1)\}$(非2的方幂数),$S_2=\{x\mid$ 有 y,z,使 $x=(y+1)(z+1)\}$(合数),$S_3=\{(x,y)\mid$ 有 u,使 $xu=y\}$(x 能整除 y),$S_4=\{(x,y)\mid$ 有 v,使 $x+v=y\}$(即 $x\leqslant y$)等集合,均为 D-集。

那么,如果 S 递归,则对任给的 $a_1,\cdots,a_n$,均可知道 $P(a_1,\cdots,a_n,y_1,\cdots,y_m)=0$ 有解与否。因为,对任给的 $a_1,\cdots,a_n$,可计算一下,看 $(a_1,\cdots,a_n)$ 是否属于 S,若属于,则有解;否则,就无解。

只要将递归集、递归可枚举集通过递归函数这一计算模型描述和讨论,则易知,任何 D-集均为递归可枚举集。而反过来,**是否任一递归可枚举集均是 D-集呢?**这便是1970年以前的三十多年里所孜孜以求的。因为一旦这个结论成立,再加上**定理34**(有递归可枚举而非递归的集合),则 Hilbert 第十问题的否定解答便已经得到。

M. Davis,J. Robinson 和 H. Putnam 在50～60年代作了大量的工作,他们已将这一结果化归到一个很局限的范围内。然而,始终未能最后解决。正如前边说,在这些工作的基础之上,Y. Matijasevič 完成了这一有决定意义的一步。

由这里可见,证明递归不可解的另一条途径是向已知的递归不可解问题化归(而**定理27**证明停机问题不可解,则是通过对角线

方法直接得到的)。这种相互化归的研究便引到递归不可解度的研究(即到底不可解的程度如何)。

经这二小节的讨论,可如图2-2表示所得结果。首先,这里所讨论的均为可数集。最内一圈表示递归集;而两个圈之间的为递归可枚举但不递归的集合,其中任一集合的补集均在外圈和方框之间。

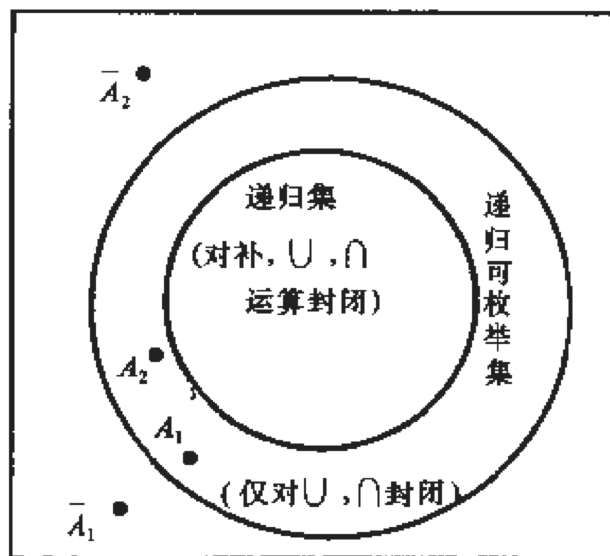


图2-2

4.7 一致可分(简记作 US)的 MMCM

本节,研究 MMCM 的子类,即所谓的一致可分的 MMCM。首先给出这种 MMCM 的定义,并证明:如果用一进制表示计算函数,则该种 MMCM 能计算任意的递归函数;如果用多进制表示计算函数,则该种 MMCM 至少可计算所有的原始递归函数。其次,简单介绍一下对更特殊的该种 MMCM 的运行过程的并行模拟。

称一个 MMCM μ 是 US 的,如果它还满足下述限制的话。设

$$e = v_1\overset{\downarrow}{a}_1 v_2\overset{\downarrow}{a}_2\cdots v_m\overset{\downarrow}{a}_m v_{m+1} \in E, \qquad q \in Q,$$

$$P^{(1)}(e,q) = (e',q'), \qquad e' \in E', q' \in Q,$$

则各限制为:

(1) e'的长度只依赖于 m 和 q(和各具体的 a_i 无关);

(2) q'也只依赖于 m 和 q；

(3) $V\cup\overset{\downarrow}{V}$中的符号，哪个在 e' 里出现以及在 e' 中的具体位置也只依赖于 m 和 q；

(4) $\overset{\downarrow}{A}$中符号在 e' 中出现的位置（仅仅是位置，这与（3）有差别）也只依赖于 m 和 q；

(5) $A\cup\overset{\downarrow}{A}$中的符号，哪个在 e' 里出现，仅由 a_1，…，a_m 和 q 决定（这一条正是由于有（3）、（4）的差别，才有）。

该定义中的限制就是说：只要 $P^{(1)}$的自变量中的 q 和 m 定了（和 e 中的其他均无关；只和 e 的长度相关），则因变量中的 q'、e' 的长度、$V\cup\overset{\downarrow}{V}$中符号在 e' 中的出现情况（哪个符号和在哪个位置出现）、$\overset{\downarrow}{A}$中符号在 e' 中出现的位置就完全确定了；而 $A\cup\overset{\downarrow}{A}$中的符号，哪个在 e' 中出现，则除取决于 m 和 q 外，还取决于自变量 e 中的 a_1，…，a_m。

Turing 机并不是一个 US MMCM。因为尽管它满足定义中的(1)，但不满足(2)～(5)。

例38 US MMCM，设 $A^{(1)}=\{a_1\}$；

$$\mu_1:(v_1\overset{\downarrow}{*},q_1)\longrightarrow(a_1\overset{\downarrow}{*},q_f);$$

$$\mu_2:(v_1\overset{\downarrow}{*},q_1)\longrightarrow(a_1v_1\overset{\downarrow}{*},q_f);$$

$$\mu_3:(v_1\overset{\downarrow}{*}v_2\overset{\downarrow}{*}\cdots v_n *,q_1)\longrightarrow(v_i\overset{\downarrow}{*},q_f),i=1,\cdots,n;$$

不难看到，如果假定 US MMCM 计算过程开始时总为 $(\underbrace{a_1\cdots a_1}_{x_1+1}\overset{\downarrow}{*}\underbrace{a_1\cdots a_1}_{x_2+1}\overset{\downarrow}{*}\cdots\underbrace{a_1\cdots a_1}_{x_n+1}\overset{\downarrow}{*},q_1)$，而输出总为 $(\underbrace{a_1\cdots a_1}_{y+1}\overset{\downarrow}{*},q_f)$，则称该 US MMCM 计算了函数 $f(x_1,\cdots,x_n)=y$。那么，上述**例38**中的 μ_1，μ_2，μ_3分别计算了“零”函数、后继函数和射影函数。

定理 35 设函数 $g_1(x_1,\cdots,x_m)$，…，$g_n(x_1,\cdots,x_m)$，$h(x_1,\cdots x_n)$均可由 US MMCM 计算，则

$$f(x_1,\cdots,x_m)=h(g_1(x_1,\cdots,x_m),\cdots,g_n(x_1,\cdots,x_m))$$

也可由 US MMCM 计算。

证明　首先，介绍一下计算 $f(x_1,\cdots,x_n)$的过程，而后再给出计算它的 US MMCM μ。

对所给的输入值

$$a=\underbrace{a_1\cdots a_1}_{x_1+1}\overset{\downarrow}{*}\cdots\overset{\downarrow}{*}\underbrace{a_1\cdots a_1}_{x_m+1}\overset{\downarrow}{*},$$

首先复制 n 遍，以备计算各 $g_i(i=1,\cdots,n)$用，这样，便得到

$$\underbrace{a\cdots a}_{n},$$

继而，对右边的 a，模拟 $g_n(x_1,\cdots,x_m)$的计算，得到

$$\underbrace{a\cdots a}_{n-1}\ \underbrace{a_1\cdots a_1}_{g_n(x_1,\cdots,x_m)+1}\overset{\downarrow}{*},$$

和

$$\underbrace{a_1\cdots a_1}_{g_n(x_1,\cdots,x_m)-1}\ \overset{\downarrow}{*}\ \underbrace{a\cdots a}_{n-1},$$

类似计算 $g_{n-1}(x_1,\cdots,x_m),\cdots,g_1(x_1,\cdots,x_m)$，得到

$$\underbrace{a_1\cdots a_1}_{g_1(x_1,\cdots,x_m)+1}\ \overset{\downarrow}{*}\cdots\overset{\downarrow}{*}\ \underbrace{a_1\cdots a_1}_{g_n(x_1,\cdots,x_m)+1}\ \overset{\downarrow}{*}。$$

最后，模拟计算 $h(x_1,\cdots,x_n)$，即得所求。

现在列出计算 $f(x_1,\cdots,x_n)$的 US MMCM μ：

(1) $(v_1\overset{\downarrow}{*}\cdots v_m\overset{\downarrow}{*},q_1)\longrightarrow(\underbrace{\underbrace{v_1\overset{\downarrow}{*}\cdots v_m\overset{\downarrow}{*}}\cdots\underbrace{v_1\overset{\downarrow}{*}\cdots v_m\overset{\downarrow}{*}}}_{n},q_1^n)$，为 μ 的指令；

(2)设计算 $g_i(x_1,\cdots,x_n)$的 US MMCM μ_i，其若有指令：

$$(e,q_u)\rightarrow(e',q_w),$$

则 μ 就有相应的指令：

$$(v_1\overset{\downarrow}{*}\cdots v_{N_i}\overset{\downarrow}{*}e,q_u^i)\rightarrow(v_1\overset{\downarrow}{*}\cdots v_{N_i}\overset{\downarrow}{*}e',q_w^i),$$

其中，$N_i=(n-i)+(i-1)m$；

(3) $(v_1\overset{\downarrow}{*}\cdots v_{N_i}\overset{\downarrow}{*}v\overset{\downarrow}{*},q_o^i)\rightarrow(v*v_1\overset{\downarrow}{*}\cdots v_{N_i}\overset{\downarrow}{*},q_1^{i-1})$也为 μ 的指令(其中，$i=1,\cdots,n$，N_i 同(2)中的)。

那么，该 μ，先使用(1)中指令，将输入抄写 n 遍；而后，即使用(2)中的指令，

$$(v_1 \overset{\downarrow}{*} \cdots \overset{\downarrow}{*} v_{(n-1)m} \overset{\downarrow}{*} \underbrace{v_{(n-1)m+1} \overset{\downarrow}{*} \cdots v_{n\cdot m} \overset{\downarrow}{*}}_{e}, q_1^n) \to$$

$$(v_1 \overset{\downarrow}{*} \cdots v_{(n-1)m} \overset{\downarrow}{*} e', q_w^n),$$

开始模拟 μ_n 的动作，而最后得到：

$$(v_1 \overset{\downarrow}{*} \cdots v_{(n-1)m} \overset{\downarrow}{*} \underbrace{a_1 \cdots a_1}_{g_n(x_1,\cdots,x_m)+1} \overset{\downarrow}{*}, q_0^n),$$

接着使用(3)中的指令，

$$(v_1 \overset{\downarrow}{*} \cdots v_{(n-1)m} \overset{\downarrow}{*} v \overset{\downarrow}{*}, q_0^n) \to (v \overset{\downarrow}{*} v_1 \overset{\downarrow}{*} \cdots v_{(n-1)m} \overset{\downarrow}{*}, q_1^{n-1}),$$

得到$(\underbrace{a_1 \cdots a_1}_{g_n(x_1,\cdots,x_m)+1} \overset{\downarrow}{*} v_1 \overset{\downarrow}{*} \cdots \overset{\downarrow}{*} v_{(n-1)m} \overset{\downarrow}{*}, q_1^{n-1})$；以后，又开始使用(2)中指令模拟 μ_{n-1}，…直至最后得到

$$(\underbrace{a_1 \cdots a_1}_{g_1(x_1,\cdots,x_m)+1} \overset{\downarrow}{*} \cdots \underbrace{a_1 \cdots a_1}_{g_n(x_1,\cdots,x_m)+1} \overset{\downarrow}{*}, q_1^0);$$

(4)设计算 $h(x_1,\cdots,x_n)$的 US MMCM μ_H 有指令

$$(e, q_u) \to (e', q_w),$$

则 μ 有相应的指令

$$(e, q_u^0) \to (e', q_w^0);$$

那么，接着在上述(1)～(3)所运行得到的结果，使用

$$(\underbrace{\underbrace{a_1 \cdots a_1}_{g_1(x_1,\cdots,x_m)+1} \overset{\downarrow}{*} \cdots \underbrace{a_1 \cdots a_1}_{g_n(x_1,\cdots,x_m)+1}}_{e} \overset{\downarrow}{*}, q_1^0) \to (e', q_w),$$

即开始模拟 μ_H 的动作，直至最后得到

$$(\underbrace{a_1 \cdots a_1}_{h(g_1,\cdots,g_n)+1} \overset{\downarrow}{*}, q_0^0),$$

停止动作。

定理35 证毕。

定理36 类似 Turing 机模拟原始递归算子，可用 US MMCM模拟原始递归算子。

证明　设 $f(x_1,\cdots,x_n)$ 为：

$$\begin{cases} f(0,x_2,\cdots,x_n)=g(x_2,\cdots,x_n) \\ f(t+1,x_2,\cdots,x_n)=h(f(t,x_2,\cdots,x_n), \\ \qquad\qquad\qquad t,x_2,\cdots,x_n)。 \end{cases}$$

那么，计算 f 的 US MMCM μ_f，其指令为：

(1) $(v_1\overset{\downarrow}{a}_1v_2\overset{\downarrow}{a}_2\cdots v_n\overset{\downarrow}{a}_n,q_1)\rightarrow$

$$(v_1\overset{\downarrow}{a}_1\overset{\downarrow}{*}v_1\overset{\downarrow}{a}_1\overset{\downarrow}{*}v_2\overset{\downarrow}{a}_2\cdots v_n\overset{\downarrow}{a}_nv_2\overset{\downarrow}{a}_2\cdots v_n\overset{\downarrow}{a}_n,q_1^1);$$

(2) 设 g 可用 US MMCM μ_g 计算，若其有指令

$(e,q_u)\rightarrow(e',q_w)$，

则 μ 有相应的指令

$$(v_1\overset{\downarrow}{a}_1v_2\overset{\downarrow}{a}_2\cdots v_N\overset{\downarrow}{a}_ne,q_u^1)\rightarrow(v_1\overset{\downarrow}{a}_1v_2\overset{\downarrow}{a}_2\cdots v_N\overset{\downarrow}{a}_Ne',q_w^1),$$

其中，$N=n+3$；

(3) $(v_1\overset{\downarrow}{a}_1v_2\overset{\downarrow}{a}_2\cdots v_N\overset{\downarrow}{a}_n,q_0^1)\rightarrow(\overset{\downarrow}{v}_1\overset{\downarrow}{v}_2v_3\overset{\downarrow}{a}_3\cdots v_N\overset{\downarrow}{a}_N,q_l)$

其中，$N=n+4$；

(4) $(v_1\overset{\downarrow}{a}_1v_2\overset{\downarrow}{a}_2\cdots v_{n+4}\overset{\downarrow}{a}_{n+4},q_l)\rightarrow(\overset{\downarrow}{v}_1\overset{\downarrow}{v}_1v_3\overset{\downarrow}{a}_3\cdots v_{n+4}\overset{\downarrow}{a}_{n+4},q_l)$；

(5) $(v_1\overset{\downarrow}{a}_1v_2\overset{\downarrow}{a}_2\cdots v_{n+3}\overset{\downarrow}{a}_{n+3},q_l)\rightarrow(v_2\overset{\downarrow}{a}_2v_3a\overset{\downarrow}{a}_3v_2\overset{\downarrow}{a}_2v_3a\ \overset{\downarrow}{a}_3v_4\overset{\downarrow}{a}_4\cdots v_{n+3}$
$\overset{\downarrow}{a}_{n+3}v_3a\ \overset{\downarrow}{a}_3v_4\overset{\downarrow}{a}_4\cdots v_{n+2}\overset{\downarrow}{a}_{n+2},q_1^2)$；

(6) $(v_1\overset{\downarrow}{a}_1v_2\cdots v_{n+2}\overset{\downarrow}{a}_{n+2},q_l)\rightarrow(v_{n+2}\overset{\downarrow}{a}_{n+2},q_f)$；

(7) 设 h 可用 US MMCM μ_h 计算，若其有指令

$$(e,q_u)\rightarrow(e',q_w),$$

则 μ 有相应的指令

$$(v_1\overset{\downarrow}{a}_1v_2\overset{\downarrow}{a}_2\cdots v_N\overset{\downarrow}{a}_Ne,q_u^2)\rightarrow(v_1\overset{\downarrow}{a}_1v_2\overset{\downarrow}{a}_2\cdots v_N\overset{\downarrow}{a}_Ne',q_w^2),$$

其中，$N=n+3$。

有兴趣的读者可以说明其运行过程，正好满足所求。

类似地有：

定理37　类似 Turing 机模拟 μ-算子，可用 US MMCM 模拟 μ-算子。

证明　设 $f(x)=\mu t(g(x,t)=0)$，那么，计算 $f(x)$ 的 US MMCM μ_f，其指令为

(1) $(v_1\overset{\downarrow}{a}_1, q_1) \to (v_1\overset{\downarrow}{a}_1 \overset{\downarrow}{*} v_1\overset{\downarrow}{a}_1 \overset{\downarrow}{*}, q_1^1)$;

(2) 设计算 $g(x,t)$ 的 US MMCM μ_g，对其任何指令

$$(e, q_n) \to (e', q_w),$$

μ_f 有相应的指令

$$(v_1\overset{\downarrow}{a}_1 v_2\overset{\downarrow}{a}_2 e, q_n^1) \to (v_1\overset{\downarrow}{a}_1 v_2\overset{\downarrow}{a}_2 e', q_w^1);$$

(3) $(v_1\overset{\downarrow}{a}_1 v_2\overset{\downarrow}{a}_1 v_3\overset{\downarrow}{a}_3, q_0^1) \to (v_1\overset{\downarrow}{a}_1 v_2\overset{\downarrow}{a}_2 \overset{\downarrow}{v}_3, q_l)$;

(4) $(v_1\overset{\downarrow}{a}_1 v_2\overset{\downarrow}{a}_2, q_l) \to (v_2\overset{\downarrow}{a}_2, q_0)$。

对于多进制表示计算函数，US MMCM 可计算原始递归函数及全定义的递归函数。有兴趣的读者请参阅陈奇的硕士毕业论文“一致可分 MMCM 的能力分析”，这里不再多作介绍。作者认为，对一致可分的 MMCM 的性质尚有待更进一步的研究。

从上述关于 US MMCM 的定义中，所谓的“一致性”是可以体会出来的，而所谓的“可分性”则很难体会了。事实上，胡国定在其研究论文“Parallel Computation Simulating Seguential Computation”中是“动态地”给出 US 的定义的。从那里的定义，一致性与可分性都是十分明白的。这里就作以简单介绍。

对任给的 MMCM μ，对任给的输入字

$$e = a_1 \cdots a_{i_1-1}\overset{\downarrow}{a}_{i_1} \cdots \overset{\downarrow}{a}_{i_m} a_{i_m+1} \cdots a_m,$$

称 $a_{i_1}, \cdots, a_{i_m}$ 为**操作符号出现**，即要对这种操作符号施行 μ 的一步运行，而 e 中其他的符号出现，如 $a_1, \cdots, a_{i_1-1}, a_{i_1+1}, \cdots, a_{i_2-1}, a_{i_2+1}, \cdots a_m$ 均为**非操作符号出现**，对它们(除了作为一个整体，即对变元 $v_1, v_2, \cdots, v_{m+1}$ 代入)则仅仅是改换位置，并无进行什么运算。从 e 施行一步运行后，得到

$$e' = a_1' \cdots a'_{j_1-1}\overset{\downarrow}{a}{}'_{j_1} \cdots \overset{\downarrow}{a}{}'_{j_n} a'_{j_n+1} \cdots a_n',$$

其中的符号出现也分成二种：一种是在 e 中出现过的，即或者是 e 中操作字符出现，而在一步运行中并未被改变，或者是在某变元代入时一起被保留下来，这种符号出现均称作**保留符号出现**；另一种则正相反，其是 e 中原来所没有出现过的，经这一步运行而新得到的，故称作**新生符号出现**。

这样，新生符号出现和保留符号出现**可分**成两部分分别得到。但是，这两种出现的位置变化，实际上也是一种操作。所以只有上述的可分性还不能得到统一的结果。这便要求：对任给的长度相同的输入字 e，经 μ-步运行后所得的结果，必须长度相同，新生符号出现的位置也相同，且下一步运行时的操作符号出现的位置也相同。这便是所谓的**一致性**。

从这种一致性和可分性，对任给的**长度为 n 的字**，μ 对它运行时，所使用的指令和次序完全是相同的，且在运行中所保留符号出现的位置也相同，新生的符号出现位置也相同。这样，便可统一进行研究，将那些能分开得到的结果，尽早分别并行地得到。胡国定在他的文章中就是把这种并行性均找出来，即所谓的并行模拟。

然而，要实现并行模拟，必须还要有相应的并行语言，用它来描述这种模拟过程。为此，作者在文章“Petri 网一类语言”中，则完全相应于 MMCM，利用 Petri 网的天生的并行描述性质，定义了三个并行语言。赵杰在他的硕士毕业论文“Simulating Uniform Separable Computation with Petri Net-like Languages”中用 Petri 网语言 PII 实现了胡国定所给的并行模拟过程。周志东的硕士论文不仅改进了赵杰的结果，而且在 PC 机上用 C-语言建立了两个系统：一个是解释执行 PII 的系统，一个是用 PII 模拟 US MMCM 并行过程的系统。从而，可在微机上表演这种模拟过程的运行。

不过，问题出现在胡国定所给的这种并行模拟不是模拟**程序**，而是一个程序运行过程，故在赵和周的文章中用 PII 所实现的过程并无重复执行任何一条指令（或通俗、直观点说，所有的运行都是不“回头”的）。因而，即使是很简单的 US MMCM μ（比如说是“乘法”），即使对长度 n 很小（比如说是4）的输入，所编写的 PII 程序也是十分之大（大约是顺序的100步），以致无法在 PC 机上进行表演。还有一个问题就是有个“长度 n”的限制，使得至少从理论上讲就不够完美了（当然，没有“长度 n”的限制时，可能无法实现并行或尽量并行的模拟。这也有待证明）。

由此，一个自然的想法是如何将上述模拟过程改成一个模拟程序（这应不难作到，特别是有了 US MMCM 的静态定义之后），另一个想法是，上述“长度 n”能否去掉，如何去掉或者证明不能去掉。要将后一想法变成现实，可能首先要作的是对 US MMCM 的性质进行更深入的研究。在某种意义上说，US MMCM 是 Post 系统的子系统。事实上，历来，对 Post 系统的子系统还甚缺乏研究。而这种研究对不少领域都是有重要意义的。

第三章　有穷性逻辑和有穷性数学

正如引言中所说，数理逻辑最早可追溯到17世纪的 Leibniz，但真正能称得起命题逻辑的，是到19世纪中期出现了 Boole 的结果，而基本上能够描述数学的逻辑，则直到1879年的 Frege 的谓词逻辑（或一阶逻辑），才算开始建立起来。此后，特别是20世纪初的第三次数学危机，使谓词逻辑发展得更为完善。1929年，Tarski 给逻辑作了语义解释，1930年 Gödel 证明了这种逻辑的完全性。但就在第二年，Gödel 指出了这种逻辑描述能力的不足，这就是著名的 Gödel 不完全性定理。

本章，先以群为例子，介绍谓词逻辑的语法、语义和推理规则。然后，全面扼要地介绍一阶逻辑，并证明完全性定理等有关结果，同时介绍 Gödel 不完全性定理，引入二阶逻辑和 Q 逻辑。再后，引入有穷性逻辑和有穷性数学，并证明相关的结果。为更清楚地介绍一般逻辑和一般数学，本章最后着重介绍无穷命题演算，且为了与有穷的相对照，还介绍了有穷命题演算。

第一节　数理逻辑和数学

数理逻辑，正如其英译名“mathematical logic”所表示的，是用数学的方法来研究数学中的逻辑。本节，就是以群为例子，来看一下数理逻辑是用怎么样的方法，又是如何研究群中定理间的逻辑的。

1.1　群的定义和一个定理

下面是群的定义和左单位元存在定理的证明，分左、右两栏，左边为代数中的叙述，右栏则为稍符号化了的叙述。

群 G 的定义：	**（对应的符号化描述）**
不空集 G 对代数运算——乘法·，作成一个群，如果满足：	0. $(Ex)G(x)$，
Ⅰ. G 对乘法封闭；	1. $\forall x,y\{G(x) \& G(y) \Rightarrow (Ez)[G(z) \& x \cdot y = z]\}$，
Ⅱ. 乘法满足结合律，即：对任 $a,b,c \in G$，$a \cdot (b \cdot c) = (a \cdot b) \cdot c$；	2. $\forall x,y,z\{G(x) \& G(y) \& G(z) \Rightarrow x \cdot (y \cdot z) = (x \cdot y) \cdot z\}$，
Ⅲ. 对任 $a,b \in G$，方程 $a \cdot x = b$ 在 G 中有解，	3a. $\forall x,y\{G(x) \& G(y) \Rightarrow (Ez)[G(z) \& x \cdot z = y]\}$，
方程 $ya = b$ 在 G 中也有解，	3b. $\forall x,y\{G(x) \& G(y) \Rightarrow (Ez)[G(z) \& z \cdot x = y]\}$.
群的一个定理	**（相应的符号化叙述）**
定理，G 里至少存在一个称作左单位元的 e，使对任 $a \in G$，有：$e \cdot a = a$.	$Th.$ $(Ew)\{G(w) \& \forall x[G(x) \Rightarrow w \cdot x = x]\}$，

证明 ∵G 不空，	∵ $(Ex)G(x)$	(1)
∴可设有 $b \in G$，	∴可设有 b，$G(b)$	(2)
由Ⅲ，方程	由(3b)，得 $G(b) \& G(b) \Rightarrow (Ez)[G(z) \& z \cdot b = b]$	(3)
$y \cdot b = b$ 有解	由(2)，有 $G(b) \& G(b)$	(4)
即在 G 中有 e，使	由(3)，(4)得 $(Ez)[G(z) \& z \cdot b = b]$	(5)
$e \cdot b = b$，	∴可设有 e，使 $G(e) \& e \cdot b = b$	(6)
现证，对 G 中任 a，有	现证，该 e，使	
$e \cdot a = a$，	$\forall x[G(x) \Rightarrow e \cdot x = x]$	(25)
由Ⅲ，对上述 b,a，	这只要，任给一 a，$G(a)$	(7)
	则有 $e \cdot a = a$	(24)

$b\cdot x=a$ 有解，	由(3a)，有 $G(b)\&G(a)\Rightarrow(Ez)$
	$[G(z)\&b\cdot z=a]$ (8)
	由(2)，(7)，有 $G(b)\&G(a)$ (9)
即可设有 C，使	故由(8)，(9)得 $(Ez)[G(z)\&b\cdot z=a]$ (10)
$b\cdot c=a$	∴可设有 c，使 $G(c)\&b\cdot c=a$ (11)
由 Ⅱ，有	对上述 e,b,c，由(2)，得
	$G(e)\&G(b)\&G(c)\Rightarrow e\cdot(b\cdot c)$
	$=(e\cdot b)\cdot c$ (12)
$e\cdot(b\cdot c)=(e\cdot b)\cdot c$，	由(6)，(2)，(11)，得
	$G(e)\&G(b)\&G(c)$ (15)
	由(15)，(12)，得 $e\cdot(b\cdot c)=(e\cdot b)\cdot c$ (16)
故 $e\cdot a=e\cdot(b\cdot c)$	由(11)，(16)，有 $e\cdot a=(e\cdot b)\cdot c$ (18)
$=(e\cdot b)\cdot c$	由(6)，(18)，有 $e\cdot a=b\cdot c$ (20)
$=b\cdot c$	由(11)，(20)，有
$=a$，	$e\cdot a=a$ (22)
证毕。	证毕。

现在，对上述右栏的符号化进一步完全用符号叙述，即不仅对公理、定理的叙述符号化，且对定理证明中所用的逻辑推理规则给予符号表示。

群的公理 $\mathfrak{G}$

G_0 $(\exists x)G(x)$，

G_1 $(\forall x)(\forall y)[G(x)\wedge G(y)\rightarrow(\exists z)(G(z)\wedge x\cdot y=z]$，

G_2 $(\forall x)(\forall y)(\forall z)[G(x)\wedge G(y)\wedge G(z)\rightarrow x\cdot(y\cdot z)=(x\cdot y)\cdot z]$，

G_3^a $(\forall x)(\forall y)[G(x)\wedge G(y)\rightarrow(\exists z)(G(z)\wedge x\cdot z=y)]$，

G_3^b $(\forall x)(\forall y)[G(x)\wedge G(y)\rightarrow(\exists z)(G(z)\wedge z\cdot x=y)]$。

群 G 的左单位元存在定理：

$\mathfrak{G}\vdash(\exists w)[G(w)\wedge(\forall x)(G(x)\rightarrow w\cdot x=x)]$，其中，“$\vdash$”表示(由 $\mathfrak{G}$)”可推导出”($(\exists w)[\cdots\cdots]$)。

在由 $\mathfrak{G}$ 推导 $(\exists w)[G(w)\wedge(\forall x)(G(x)\to w\cdot x=x)]$ 时，用到了下述一些推理规则：

$(\in)$ $A_1,\cdots,A_n\vdash A_i(i=1,\cdots,n)$； （属于规则）

(τ) 若 $\Gamma\vdash\triangle$，且 $\triangle\vdash A$，

则 $\Gamma\vdash A$； （传递规则）

$(\wedge_+)$ $A,B\vdash A\wedge B$； （合取引入规则）

$(\wedge_-)$ $A\wedge B\vdash A,B$； （合取消去规则）

$(\to_+)$ 若 $\Gamma,A\vdash B$，则 $\Gamma\vdash A\to B$； （蕴涵引入规则）

$(\to_-)$ $A,A\to B\vdash B$； （蕴涵消去规则）

$(\exists_+)$ $A(a)\vdash(\exists x)A(x)$ （存在引入规则）

$(\exists_-)$ 若 $A(a)\vdash B$，且 B 中无 a 出现，

则 $(\exists x)A(x)\vdash B$； （存在消去规则）

$(\forall_+)$ 若 $\Gamma\vdash A(a)$，且 Γ 中无 a 出现，

则 $\Gamma\vdash(\forall x)A(x)$； （全称引入规则）

$(\forall_-)$ $(\forall x)A(x)\vdash A(a)$； （全称消去规则）

(I_-) $A(a),a=b\vdash A(b)$； （等词消去规则）

此外，再增加两条上述推导中没用到的规则：

(I_+) $\vdash a=a$； （等词引入规则）

$(\neg)$ 若 $\Gamma,\neg A\vdash B,\neg B$，

则 $\Gamma\vdash A$； （反证法规则）

上述13条推理规则中，除 $\Gamma,\triangle$ 是数学公式的有穷序列外，而其他如 $A,B,A(a),(\forall x)A(x),(\exists x)A(x),A_i(i=1,\cdots,n),a=b,A(b)$ 等都是一个个的数学公式。其他一些更细微的地方下节再详细、严格地定义。另外，这些公式和公式序列都是泛指的，如 A，即表示任意一个公式，Γ 则表示任意一个公式序列。这就是所谓的元语言记号，这下面再介绍。

可以这样说，各推理规则实际上是定义了数学公式序列和数学公式间的一个二元关系；且有些规则是直接告诉怎样的序列和怎样的公式间有这种二元的推导关系，而有些规则则是间接告诉的。如 $(\in)$ 当然是直接告诉的；而 (τ) 则是间接告诉的，即如果已知

有 $\Gamma \vdash \triangle$（这实际上，是 $\Gamma \vdash \delta_1, \cdots, \Gamma \vdash \delta_m$ 的缩写，而$\triangle$即 $\delta_1, \cdots, \delta_m$），且还已知有$\triangle \vdash A$，那么才有 $\Gamma \vdash A$。又如$(\exists_-)$，其自然是间接告诉的，且还有个附加条件"B 中无 a 出现"。

应该说，这些规则，像 Turing 机的一条指令一样，是非常简单的，且更重要的是有个明确的特点：完全从**形式上能立即可判定**规则的成立与否。

现在，把上述定理的证明过程完全按上述推理规则一步步列下，且所标出的步号，与前述符号化的叙述相一致。

(0) $\mathfrak{G}$

(1) $(\exists x)G(x)$ $\quad$ ((0),($\in$))

(2) $\quad G(b)$ $\quad$ (直至推导出(29))

(3) $\quad G(b) \wedge G(b) \rightarrow (\exists z)(G(z) \wedge z \cdot b = b)$ $\quad$ (G_3^b，二次($\forall_-$))

(4) $\quad G(b) \wedge G(b)$ $\quad$ ((2),($\wedge_+$))

(5) $\quad (\exists z)(G(z) \wedge z \cdot b = b)$ $\quad$ ((4),(3),($\rightarrow_-$))

(6) $\quad\quad G(e) \wedge e \cdot b = b$ $\quad$ (直至推导出(28))

(7) $\quad\quad\quad G(a)$ $\quad$ (直至推导出(23))

(8) $\quad\quad\quad G(b) \wedge G(a) \rightarrow (\exists z)(G(z) \wedge b \cdot z = a)$

$\quad$ (A_3^a，二次($\forall_-$))

(9) $\quad\quad\quad G(b) \wedge G(a)$ $\quad$ ((2),(7),($\wedge_+$))

(10) $\quad\quad\quad (\exists z)(G(z) \wedge b \cdot z = a)$ $\quad$ ((9),(8),($\rightarrow_-$))

(11) $\quad\quad\quad\quad G(c) \wedge b \cdot c = a$ $\quad$ (直至推导出(22))

(12) $\quad\quad\quad\quad G(e) \wedge G(b) \wedge G(c) \rightarrow e \cdot (b \cdot c) = (e \cdot b) \cdot c$

$\quad$ (G_2，三次($\forall_-$))

(13) $\quad\quad\quad\quad G(e)$ $\quad$ ((6),($\wedge_-$))

(14) $\quad\quad\quad\quad G(c)$ $\quad$ ((11),($\wedge_-$))

(15) $\quad\quad\quad\quad G(e) \wedge G(b) \wedge G(c)$

$\quad$ ((13),(2),(4)，二次($\wedge_+$))

(16) $\quad\quad\quad\quad e \cdot (b \cdot c) = (e \cdot b) \cdot c$ $\quad$ ((15),(12),($\rightarrow_-$))

(17) $\quad\quad\quad\quad b \cdot c = a$ $\quad$ ((11),($\wedge_-$))

(18) $\quad\quad\quad\quad e \cdot a = (e \cdot b) \cdot c$ $\quad$ ((16),(17),(I_-))

(19) $e \cdot b = b$ ((6),($\wedge_-$))

(20) $e \cdot a = b \cdot c$ ((18),(19)(I_-))

(21) $b \cdot c = a$ ((11),($\wedge$))

(22) $e \cdot a = a$ ((20)(21),(I_-))

(23) $e \cdot a = a$ ((11)(22),(10),($\exists_-$))

(24) $G(a) \rightarrow e \cdot a = a$ ((7),(23),($\rightarrow_+$))

(25) $(\forall x)(G(x) \rightarrow e \cdot x = x)$ ((24),($\forall_+$))

(26) $G(e)$ ((6),($\wedge_-$))

(27) $G(e) \wedge (\forall x)(G(x) \rightarrow e \cdot x = x)$

((26)(25),($\wedge_+$))

(28) $(\exists w)[G(w) \wedge (\forall x)(G(x) \rightarrow w \cdot x = x)]$

((27)($\exists_+$))

(29) $(\exists w)[G(w) \wedge (\forall x)(G(x) \rightarrow w \cdot x = x)]$

((6),(28),(5),($\exists_-$))

(30) $(\exists w)[G(w) \wedge (\forall x)(G(x) \rightarrow w \cdot x = x)]$

((2),(29),(1),($\exists_-$))。

该种书写方式为“斜形证明”(参见《数理逻辑基础》第61～65页)。

1.2 语言和推理规则

从上一小节的群的例子可见,数理逻辑所用的数学方法,就是用以描述数学的**语言**,以及从数学定理证明中所提炼出的**推理规则**。这两者,恰与理想计算机语言中的两部分内容(描述运算对象和对计算装置进行操作的指令)相对应着。

推理规则已如上小节介绍,有关它们的更进一步讨论在下一小节中进行。这里仅对语言加以讨论。

个体词:a,b,c,d,e 等,或加下标;(其表示群的任一给定的元素的)

变元:x,y,z,w 等,或加下标;(其表示群的任意元素的)

谓词:一元谓词 G,二元谓词 $=$;(它们分别表示群的所有元素集合和群的相等元素的)

函数词：二元运算·；（表示群的乘法运算）

逻辑连接词：∧（合取），→（蕴涵），¬（非），∀（全称量词），∃（存在量词）；

辅助符号：(，)（左、右括弧）。

以上六类符号即为该语言的符号。现在介绍如何用符号来构造更大的语言单元的。

项：个体词，变元都是项；且若 t_1 是项，t_2 也是项，则 $(t_1 \cdot t_2)$ 是项；

合式公式：

(1) t 是无变元出现的项，则 $G(t)$ 是合式公式；t_1，t_2 均是无变元的项，则 $t_1 = t_2$ 是合式公式；

(2) F_1，F_2 均是合式公式，则 $(F_1 \wedge F_2)$，$(F_1 \rightarrow F_2)$，$(\neg F_1)$ 均是合式公式；$F(a)$ 是合式公式，且 a 是在其中出现的个体词，则 $(\forall x) F(x)$，$(\exists x)F(x)$ 均是合式公式，其中 $F(x)$ 是由 $F(a)$ 将其中所有 a 的出现均代入为 x 的出现得到。

由上述，不难知道，前面所出现的数学公式均为合式公式。比如，在证明的第25步出现的公式 $(\forall x)(G(x) \rightarrow e \cdot x = x)$ 是合式公式，其形成过程为：$G(a)$，

$(e \cdot a) = a$，

$(G(a) \rightarrow (e \cdot a) = a)$，

$(\forall x)(G(x) \rightarrow (e \cdot x) = x)$，

所不同的只是辅助符号可能有所省略。

最后说一句，上述这种对语言的描述都是所谓的元语言描述，因为这是在介绍另外一种语言。请读者注意这一点。

1.3 数理逻辑和数学

上一小节介绍语言的同时，已将它们的涵义均作了交待。实际上，也可不给出这种涵义，而所定义的完全是一种形式符号或形式符号的有穷序列，故有时也称作形式语言。但总得将它们和数学建立联系，以便表述数学中的元素、公式和其他概念等等。这才能称

作是“研究数学的内容”。

这里，仅对1.1中的推理规则再作一研究。因为这些规则要能反映数学定理证明的推理，故自然要有两个问题必须解决：即推理的**正确性**（或称可靠性）和**完全性**。前者是指，这些规则不能太强，不能将错误的东西也推出来；后者则是指，这些规则的能力也不能太弱，使在数学中能推理得出的东西，而这里却得不到。

首先，上述规则的正确性比较明显，兹仅解释一下$(\exists_+)$和$(\exists_-)$。其实，$(\exists_+)A(a)\vdash(\exists x)A(x)$的正确性并不难理解，$A(a)$表示“个体$a$有$A$性质，或$a$属于集合$A$”，那自然可得到“有$x$，使$x$具有$A$性质”；而$(\exists_-)$是说，如果已经有

$$\Gamma, A(a)\vdash B,\text{且 } B \text{ 中无 } a \text{ 出现},$$

那就是说，B或者从Γ得到，或者从Γ和$A(a)$得到；若是前者，自然会有$\Gamma,(\exists x)A(x)\vdash B$；若是后者，再考虑到“$B$中无$a$出现”，即是说，$B$和$A(a)$中出现的“$a$”是什么并无关系，故只要“有$x$，满足$A$性质”即可推得$B$，而这就是

$$\Gamma,(\exists x)A(x)\vdash B。$$

因而，可见$(\exists_-)$的正确性。

其次，关于完全性问题，则是个不简单的问题了。因为究竟是否将数学定理证明中的推理均给予了刻画，这需要作更深入的探讨。事实上，从 Frege 建立谓词逻辑，一直到1929年，都无从谈起如何才算完全了。只有当 Tarski 给作了全面的语义解释后，才给完全性建立了一个合适的标准，而 Gödel 在第二年即完成了完全性定理的证明。这是 Gödel 的博士论文中所给出的。下一节，即会介绍这一非常重要而深刻的结果。

第二节　一阶逻辑

本节扼要、全面地介绍一阶逻辑。这包括三方面内容：即一阶逻辑语言，一阶逻辑语言的语义、及一阶逻辑的推导系统。并将证明这推导系统是正确的、完全的。然后还指出一阶逻辑语言的不足

之处。

2.1 一阶语言

一个非空的符号集合称作**至多可数的**,如果它有穷或者可数。那么易证

定理1 设符号集 M 非空,则下述(a),(b),(c)相互等价:

(a) M 至多可数;

(b) 有一个满射 $\alpha: N \to M$,其中,$N=\{0,1,2\cdots\}$;

(c) 有内射 $\beta: M \to N$,N 同上。

该定理的证明留作练习。

定理1a 若 A 至多可数,则 $A^*=\{\alpha|\alpha$ 为 A 中符号的有穷串$\}$可数。(留作练习)

一阶语言的**符号集合**

$A_S=A\cup S$,

其中,A 称作**逻辑符号集**,S 称作**非逻辑符号集**。A 包括下述(a)、(b)、(c)、(d)四类符号,S 包括下述(e),(f)、(g)三类符号。

(a) 变元符号:$v_0, v_1, v_2, \cdots$;

(b) 逻辑连接词:$\neg$,$\wedge$,$\vee$,$\to$,$\leftrightarrow$,$\exists$,$\forall$(分别称作非、合取、析取、蕴涵、等值、存在量词、全称量词);

(c) 等词符号(二元):$\equiv$;

(d) 辅助符号:(,)(左、右括弧);

(e) 谓词符号:对 $n \geqslant 1$,有 n 元谓词集合(可为空集);

(f) 函数词符号:对 $n \geqslant 1$,有 n 元函数符号集合(可为空集);

(g) 个体词集合。

常用 P、Q、R 或加下标表示谓词符号,用 f,g,h 或加下标表示函数词符号,用 a,b,c,d,e 或加下标表示个体词符号,用 x,y,z,w 或加下标表示变元符号。

现在,利用 A_S 中的符号构造更大的语言成分。

S-项集是 A_S^* 的子集 T^S,由下述几条构造:

(T_1) 变元符号是 S-项;

(T_2) 个体词符号是 S -项；

(T_3) 若 $t_0,\cdots,t_{n-1}$是 S -项，且 f 是 n 元函数词符号，则 $ft_0t_1\cdots t_{n-1}$是 S -项。

S -公式集是A_S^* 的子集 L^S，由下述几条构造：

(F_1) 若 $t_i(i=0,1)$是无变元符号出现的 S -项，则 $t_0\equiv t_1$是 S -公式；

(F_2) 若 $t_i(i=0,1,\cdots,n-1)$是无变元符号出现的 S -项，且 R 是 n -元谓词符号，则 $Rt_0t_1\cdots t_{n-1}$是 S -公式；

(F_3) 若 φ_1,φ_2是 S -公式，则$\neg\varphi_1,(\varphi_1\wedge\varphi_2),(\varphi_1\vee\varphi_2),(\varphi_1\rightarrow\varphi_2),(\varphi_1\leftrightarrow\varphi_2)$均是 S -公式；

(F_4) 若 $\varphi(a)$是 S -公式，且个体词 a 在 $\varphi(a)$中出现，变元 x 不在 $\varphi(a)$中出现，则$\forall x\varphi(x)$，$\exists x\varphi(x)$均为 S -公式。

特别称由(F_1)和(F_2)构造的 S -公式为**原子公式**。**一阶语言**即指 S -**公式集合** L^S。

例1 设 $S=\{a,b,P,Q,f,g\}$，则

$$\forall v_0(Pa\rightarrow Qv_0b),$$

$$(Pfagb\rightarrow\exists v_0(v_0\equiv v_0\wedge v_0\equiv v_0)),$$

$$\forall v_0\forall v_1\forall v_2(Qv_0v_1\wedge Pv_2)$$

均是 S -公式。(注意：P,g 是一元的，而 Q,f 是二元的)

在后面讨论中，如不是特指某非逻辑符号集 S 时，常省掉 S -项、S -公式中的"S -"，而直接称为**项、公式**。

上述(T_1)～(T_3)是由简单的项构造复杂的项，而反过来，当由复杂的项往构成它的简单的项分解时，会否得到不同的分解。或者说，同一个复杂的项，会否由两种不同的构造方式获得。下述引理和定理说明不会出现这种情况。

引理2 对任何的项 t 和 t'，不会有其他的非空字 $\zeta\in A_S^*$，使 $t\zeta=t'$，也即 t 不会是 t' 的真字首。

证明 施归纳于项 t 的构造(T_1)—(T_3)，证明 t 不是任何项 t' 的真字首，反之亦然。只要完成这样的证明，则**引理2**证毕。如下：

$t=x$：设 t' 是任意的项，则显然有，t'不是 x 的真字首(除非

$t'=\Lambda$，而 t' 不是项了)；而另方面，可施归纳于 t' 的构造，t 也不可能是 t' 的真字首。

$t=a$：类似 $t=x$ 的证明。

$t=ft_0\cdots t_{n-1}$，且对 $t_0,\cdots,t_{n-1}$ 均无 $t_0',\cdots,t'_{n-1}$，使 t_i 是 t_i' 的真字首或 t_i' 是 t_i 的真字首($i=0,1,\cdots,n-1$)；设 t' 是任给定的项，那么，t 和 t' 也互不为真字首；因为，否则，t' 必以此 f 为字首，即可设 $t'=ft_0't_1'\cdots t'_{n-1}$，且或者有非空的 ζ，使 $t=t'\zeta$，或者有非空的 ζ，使 $t\zeta=t'$，若为前者，即有，$ft_0\cdots t_{n-1}=ft_0't_1'\cdots t'_{n-1}\zeta$，消去 f，有

$$t_0\cdots t_{n-1}=t_0't_1'\cdots t'_{n-1}\zeta,$$

那么，或者 t_0 为 t_0' 的真字首，或者 t_0' 为 t_0 的真字首，而这均与归纳没矛盾，故只能是 $t_0=t_0'$，消去 t_0 和 t_0'，得

$$t_1\cdots t_{n-1}=t_1'\cdots t'_{n-1}\zeta,$$

类似地，消去 t_1 和 t_1'，t_2 和 t_2'，$\cdots$，t_{n-1} 和 t'_{n-1}，最后得 $\Lambda=\zeta$，这与 ζ 是非空的字相矛盾。类似可证，不可能有非空的 ζ，使 $t\zeta=t'$。故归纳毕，引理证毕。

定理3　若 $t_0,\cdots,t_{n-1},t_0',\cdots,t'_{m-1}$ 均是项，且

$$t_0\cdots t_{n-1}=t_0'\cdots t'_{m-1},$$

则有 $n=m$，且 $t_i=t_i'$($i<n$)。

证明　由**引理2**立得。

由**定理3**，任给项 t，其构造过程是唯一确定的。

类似地，可陈述有关公式唯一分解定理，试列出并证明之(留作练习)。

由唯一分解定理，对项或公式，只要按(T_1)－(T_3)，或(F_1)－(F_4)，对它们作了某种定义，则该种定义即是合理的，即对任一给定的项或公式，有唯一的该种定义。比如，对项，若对变元符号，对个体词符号均赋予了相应的值，且又假设对 $t_0,\cdots,t_{n-1}$ 已赋了值，对 $ft_0\cdots t_{n-1}$ 也赋了某值的话，则对任项，均赋予了值。

2.2　一阶逻辑语言的语义

在2.1中，所定义的项、公式均为符号的有穷串，尚未给其赋予

任何的涵义。本节，则是要给这些**形式对象**，赋予涵义，即为论域中的对象、论域上的关系或命题。

S-**结构**是一个二元组

$$\mathfrak{A} = (A, \alpha),$$

其中，A 是非空集，称作 $\mathfrak{A}$ 的论域，α 是 S 上的映射，其将谓词符号 R，映射为 A 上的关系 $\alpha(R)$，将函数词符号 f，映射为 A 上的函数 $\alpha(f)$，且元数也都相应着；该 α 还将个体词 a 映射为 A 中的元素 $\alpha(a)$。且常用 $R^{\mathfrak{A}}$、$f^{\mathfrak{A}}$，$a^{\mathfrak{A}}$ 来记它们，甚至用 R^A，f^A，a^A 记之。且 $\{R,f,g,a\}$-结构 $\mathfrak{A}$ 常记作 $\mathfrak{A}=(A,R^{\mathfrak{A}}f^{\mathfrak{A}},g^{\mathfrak{A}},a^{\mathfrak{A}})$ 等等。

例2 在研究群时，所给的 S 为

$$S_{gr} = \{\cdot, G\},$$

那么，S_{gr}-结构 $\mathfrak{A}=(A,\cdot^A,G^A)$，其中，$A$ 为群的元素集。这样，就将谓词符号，个体词符号，函数词符号与数学中的关系、函数、论域中元素分别建立了对应。

为了将形式对象 S-公式和数学中的命题建立对应关系，还要给出在 **S-结构 $\mathfrak{A}$ 中的赋值**，**S-解释**及**可满足关系**的定义等。

称映射 $\beta:\{v_n|n\in N\}\to A$ 为 **S-结构 $\mathfrak{A}$ 中的赋值**，即从变元符号集合到论域 A 的映射。

称序对 $\mathfrak{I}=(\mathfrak{A},\beta)$ 为 **S-解释**，其将任何 S-项都解释为 A 中的元素，即

(T_1') 对变元符号 x，$\mathfrak{I}(x)=\beta(x)$；

(T_2') 对个体词符号 a，$\mathfrak{I}(a)=a^{\mathfrak{A}}$（或 a^A）；

(T_3') 对函数词符号 f，对任给的 S-项 $t_0,\cdots,t_{n-1}$，$\mathfrak{I}(ft_0\cdots t_{n-1})=f^{\mathfrak{A}}(\mathfrak{I}(t_0),\cdots,\mathfrak{I}(t_{n-1}))$。

对 S-赋值 β，S-解释 $\mathfrak{I}$，也常简称作赋值 β，解释 $\mathfrak{I}$。

称**解释 $\mathfrak{I}$ 是公式 φ 的模型**，或称 **$\mathfrak{I}$ 满足** φ（即在解释 $\mathfrak{I}$ 之下 φ 成立），记作 $\mathfrak{I}\models\varphi$，如下归纳定义之：

(F_1') $\mathfrak{I}\models t_0\equiv t_1$ 当且仅当 $\mathfrak{I}(t_0)\equiv\mathfrak{I}(t_1)$；

(F_2') $\mathfrak{I}\models Rt_0\cdots t_{n-1}$ 当且仅当 $R^{\mathfrak{A}}(\mathfrak{I}(t_0),\cdots\mathfrak{I}(t_{n-1}))$；

(F_3') $\mathfrak{I}\models\neg\varphi_1$ 当且仅当 没有 $\mathfrak{I}\models\varphi_1$；

$\mathfrak{I}\models(\varphi_1\wedge\varphi_2)$ 当且仅当 $\mathfrak{I}\models\varphi_1$ 且 $\mathfrak{I}\models\varphi_2$；

$\mathfrak{I}\models(\varphi_1\vee\varphi_2)$ 当且仅当 $\mathfrak{I}\models\varphi_1$ 或 $\mathfrak{I}\models\varphi_2$；

$\mathfrak{I}\models(\varphi_1\rightarrow\varphi_2)$ 当且仅当 如果 $\mathfrak{I}\models\varphi_1$，则有 $\mathfrak{I}\models\varphi_2$；

$\mathfrak{I}\models(\varphi_1\leftrightarrow\varphi_2)$ 当且仅当 如果 $\mathfrak{I}\models\varphi_1$，则有 $\mathfrak{I}\models\varphi_2$，而且如果 $\mathfrak{I}\models\varphi_2$，则有 $\mathfrak{I}\models\varphi_1$；

$(F_4{}')\ \mathfrak{I}\models\forall x\varphi(x)$ 当且仅当 对所有 $a\in A$，$\mathfrak{I}\frac{a}{x}\models\varphi(x)$；

$\mathfrak{I}\models\exists x\varphi(x)$ 当且仅当 有 $a\in A$，使 $\mathfrak{I}\frac{a}{x}\models\varphi(x)$；

其中，$\mathfrak{I}\frac{a}{x}=(\mathfrak{A},\beta\frac{a}{x})$，而 $\beta\frac{a}{x}(y)=\begin{cases}\beta(y)，当\ y\neq x\ 时，\\ a，当\ y=x\ 时。\end{cases}$

称解释 $\mathfrak{I}$ 是公式集合 Φ 的模型，记作 $\mathfrak{I}\models\Phi$，如果对任 $\varphi\in\Phi$，均有 $\mathfrak{I}\models\varphi$。

称公式 φ 是公式集合 Φ 的后承，(记作 $\Phi\models\varphi$)，如果对任意解释 $\mathfrak{I}$，若有 $\mathfrak{I}\models\Phi$，则有 $\mathfrak{I}\models\varphi$。

例3 对第一节所给的群的例子，重新写一下其公理：此时 $S=\{\cdot\}$

$$G_1=\forall v_0\forall v_1\exists v_2(\cdot v_0v_1=v_2),$$

$$G_2=\forall v_0\forall v_1\forall v_2(\cdot v_0\cdot v_1v_2=\cdot\cdot v_0v_1v_2),$$

$$G_3=\forall v_0\forall v_1\exists v_2(\cdot v_0v_2=v_1),$$

$$G_4=\forall v_0\forall v_1\exists v_2(\cdot v_2v_0=v_1);$$

而左单位元存在定理为公式

$$G_5=\exists v_0\forall v_1(\cdot v_0v_1=v_1)。$$

那么，从群论的知识，知道：

$$\mathfrak{G}\models G_5，其中\ \mathfrak{G}=\{G_1,G_2,G_3,G_4\};$$

但是，既无

$$\mathfrak{G}\models\forall v_0\forall v_1(\cdot v_0v_1=\cdot v_1v_0),$$

也无

$$\mathfrak{G}\models\neg\forall v_0\forall v_1(\cdot v_0v_1=\cdot v_1v_0)。$$

由之可见，对公式集Φ，公式φ，可能下述二者均不成立：

$$\Phi \models \varphi \qquad 和\ \Phi \models \neg \varphi。$$

称公式φ**有效**，记作$\models \varphi$，如果有$\varnothing \models \varphi$。

称公式φ**可满足**，如果φ有模型$\mathfrak{I}$的话；记作Satφ；称**公式集Φ可满足**，如果有一模型$\mathfrak{I}$，对任$\varphi \in \Phi$，均有$\mathfrak{I} \models \varphi$；记作Sat Φ。

定理4 对任公式集Φ和公式φ，有：

$\Phi \models \varphi$ 当且仅当 没有Sat$(\Phi \cup \{\neg \varphi\})$。

证明 $\Phi \models \varphi$ 当且仅当 对任何解释$\mathfrak{I}$，使如果$\mathfrak{I} \models \Phi$，则有$\mathfrak{I} \models \varphi$，

当且仅当 没有解释$\mathfrak{I}$、使$\mathfrak{I} \models \Phi$，且没有$\mathfrak{I} \models \varphi$；

当且仅当 没有解释$\mathfrak{I}$，使$\mathfrak{I} \models \Phi \cup \{\neg \varphi\}$；

当且仅当 没有Sat$(\Phi \cup \{\neg \varphi\})$。

例4 等价关系可如下刻画：$S = \{R\}$，

$$\forall v_0 R v_0 v_0, \forall v_0 \forall v_1 (R v_0 v_1 \rightarrow R v_1 v_0),$$

$$\forall v_0 \forall v_1 \forall v_2 ((R v_0 v_1 \wedge R v_1 v_2) \rightarrow R v_0 v_2)。$$

设这三个公式的集合记为Φ，则有

$$\Phi \models \varphi, \qquad \varphi = \forall v_0 \forall v_1 (\exists v_2 (R v_0 v_2 \wedge R v_1 v_2) \rightarrow \forall v_2 (R v_0 v_2 \leftrightarrow R v_1 v_2))。$$

例5 设$S = \varnothing$，那么

$\mathfrak{I} = (\mathfrak{A}, \beta) \models \varphi_{\geq 2}$， $\varphi_{\geq 2} = \exists v_0 \exists v_1 \neg v_0 = v_1$

当且仅当A至少有二个元素；类似地，有$\varphi_{\geq n}$；那么

$\mathfrak{I} = (\mathfrak{A}, \beta) \models \neg \varphi_{\geq n}$ 当且仅当A有$<n$个元素；

而 $\mathfrak{I} = (\mathfrak{A}, \beta) \models \varphi_{\geq n} \wedge \neg \varphi_{\geq n+1}$ 当且仅当A恰有n个元素；

而 $\mathfrak{I} = (\mathfrak{A}, \beta) \models \Phi_{\infty}$， $\Phi_{\infty} = \{\varphi_{\geq n} \mid n \geq 2, n \in N\}$ 当且仅当A有无穷多元素。

例6 全序可用下述公式刻画：$S = \{<\}$，

$$\Phi_{\text{ord}} \begin{cases} \forall v_0 \neg v_0 < v_0, \\ \forall v_0 \forall v_1 \forall v_2 ((v_0 < v_1 \wedge v_1 < v_2) \rightarrow v_0 < v_2), \\ \forall v_0 \forall v_1 (v_0 < v_1 \vee v_0 \equiv v_1 \vee v_1 < v_0); \end{cases}$$

那么，$\mathfrak{J}=(\mathfrak{A},\beta)\models\Phi_{\mathrm{ord}}$ 当且仅当 $\mathfrak{A}=(A,<^{\mathfrak{A}})$为全序。

例7 除孤立点外的全序可刻画为：$S=\{<\}$

$$\Phi_{\mathrm{pord}}\begin{cases}\forall v_0\neg v_0<v_0, \\ \forall v_0\forall v_1\forall v_2((v_0<v_1\wedge v_1<v_2)\rightarrow v_0<v_2), \\ \forall v_0\forall v_1((\exists v_2(v_0<v_2\vee v_2<v_0)\wedge\exists v_2(v_1<v_2 \\ \qquad\vee v_2<v_1))\rightarrow(v_0<v_1\vee v_0\equiv v_1\vee v_1<v_0));\end{cases}$$

那么，$\mathfrak{J}=(\mathfrak{A},\beta)\models\Phi_{\mathrm{pord}}$ 当且仅当 $\mathfrak{A}=(A,<^{\mathfrak{A}})$为除孤立点外均全序的结构。

例8 自然数的算术可由下述公式刻画：$S=\{+,\cdot,0,1\}$，

$$\Phi_{ar}^{\mathrm{I}}\begin{cases}\forall v_0\neg v_0+1\equiv 0, \\ \forall v_0\forall v_1(v_0+1\equiv v_1+1\rightarrow v_0\equiv v_1), \\ \forall X((X0\wedge\forall v_0(Xv_0\rightarrow Xv_0+1))\rightarrow\forall v_0Xv_0), \\ \forall v_0v_0+0\equiv v_0, \\ \forall v_0\forall v_1v_0+(v_1+1)\equiv(v_0+v_1)+1, \\ \forall v_0v_0\cdot 0\equiv 0, \\ \forall v_0\forall v_1v_0\cdot(v_1+1)\equiv(v_0\cdot v_1)+v_0;\end{cases}$$

注意，上述第三个公式是二阶的，因为有谓词变元“X”。**试说明满足 Φ_{ar}^{I} 的解释 $\mathfrak{J}=(\mathfrak{A},\beta)$，其论域 A 只可能是** N(或与 N 相同构)。如完全用一阶公式写上述公式，则第三个公式就要变成可数条公式，即：对任何有一个个体词 a 在其中出现的公式 Fa，有公式

$$Fo\wedge\forall v_0(Fv_0\rightarrow Fv_0+1)\rightarrow\forall v_0Fv_0,$$

这种公式显然有可数个。试说明：**这可数个公式也抵不上第三个公式的刻划能力**(即满足第三个公式的结构一定满足这可数个公式，反之，却不一定)。事实上，也正是由此，才产生了 Gödel 不完全性定理。

2.3 一阶逻辑的推理系统

一阶逻辑推理系统，包括2.1中所定义的一阶语言 L^S，且更重要的是还包括**推理规则**和**形式证明**。本节，还要引入有关“∨”和

"$\leftrightarrow$"的四条推理规则，并进一步重述1.1中有关谓词的推理规则。所谓**形式证明**是由公式序列 Γ_i 和公式 $A_i(i=1,\cdots,n)$所组成的序对构成的有穷序列

$$(\Gamma_1,A_1)(\Gamma_2,A_2)\cdots(\Gamma_n,A_n),$$

且满足：Γ_1和 A_1根据直接告诉的推理规则，具有形式推导关系（记作 $\Gamma_1\vdash A_1$），而对任何$1<i\leqslant n$，或者是根据直接告诉的推理规则，Γ_i 和 A_i 具有形式推导关系（记作 $\Gamma_i\vdash A_i$），或者有$1\leqslant i_1<i_2<\cdots<i_m<i$，根据间接告诉的推理规则，从 $\Gamma_{i_1}\vdash A_{i_1},\cdots,\Gamma_{i_m}\vdash A_{i_m}$得到 Γ_i 和 A_i 具有形式推导关系（记作 $\Gamma_i\vdash A_i$）。此时，特称**该形式证明为 $\Gamma_n\vdash A_n$ 的形式证明**。可将其重新写作 $\Gamma_1\vdash A_1,\Gamma_2\vdash A_2,\cdots,\Gamma_n\vdash A_n$。并称 **$n$ 为该形式证明的长度**。且对其中的每个形式推导关系 $\Gamma_i\vdash A_i(i=1,\cdots,n)$，称 **$\Gamma_i$ 为 A_i 的形式前提，而 A_i 为 Γ_i 的形式结论**。

在1.1中介绍的推理规则($\in$)、(τ)、($\wedge_+$)、($\wedge_-$)、($\rightarrow_+$)、($\rightarrow_-$)、($\neg$)，这里就不再重述。兹再引入：

($\vee_+$) $A\vdash A\vee B$；　$B\vdash A\vee B$；　(析取引入)

($\vee_-$) 若 $A\vdash C$，且 $B\vdash C$，则 $A\vee B\vdash C$；　(析取消去)

($\leftrightarrow_+$) 若 $\Gamma,A\vdash B$，且 $\Gamma,B\vdash A$；则 $\Gamma\vdash A\leftrightarrow B$；(等值引入)

($\leftrightarrow_-$) $A,A\leftrightarrow B\vdash B$；　$B,A\leftrightarrow B\vdash A$；　(等值消去)

并进一步重述1.1中介绍过的有关量词和等词的六条规则，只是将那里的个体词 a,b 可能要改成项。如下：

($\exists_+$) $A(t)\vdash\exists xA(x)$，　t 为项，而 $A(x)$为将 $A(t)$中某些 t 换成 x 而得；　(存在引入)

($\exists_-$) 若 $A(a)\vdash B$，且 B 中无 a 出现，则有$\exists xA(x)\vdash B$；　(存在消去)

($\forall_+$) 若 $\Gamma\vdash A(a)$，且 Γ 中无 a 出现，则有 $\Gamma\vdash\forall xA(x)$；　(全称引入)

($\forall_-$) $\forall xA(x)\vdash A(t)$，　t 为无变元的项；　(全称消去)

(I_+)　$\vdash t=t$；　t 为无变元的项；　(等词引入)

(I_-)　$A(t_1),t_1\equiv t_2\vdash A(t_2)$，　t_1,t_2为项，$A(t_2)$为将 $A(t_1)$中

某些t_1换成t_2得到；（等词消去）

现在举一简例说明如何应用推理规则，得到一系列形式推导关系，从而构成一个形式证明，并进一步说明其相应的斜形证明是如何得到的。

例9 试应用各规则判定有形式推导：

$$A\to(B\to C),A\to B\vdash A\to C。$$

分如下10步：

(1) $A\to(B\to C),A\to B,A\vdash A\to(B\to C)$ (∈)

(2) $A\to(B\to C),A\to B,A\vdash A\to B$ (∈)

(3) $A\to(B\to C),A\to B,A\vdash A$ (∈)

(4) $A\to(B\to C),A\vdash B\to C$ $(\to_-)$

(5) $A\to(B\to C),A\to B,A\vdash B\to C$ (1)(3)(4)(τ)

(6) $A\to B,A\vdash B$ $(\to_-)$

(7) $A\to(B\to C),A\to B,A\vdash B$ (2)(3)(6)(τ)

(8) $B\to C,B\vdash C$ $(\to_-)$

(9) $A\to(B\to C),A\to B,A\vdash C$ (5)(7)(8)(τ)

(10) $A\to(B\to C),A\to B\vdash A\to C$ (9)$(\to_+)$

这从(1)—(10)，每个均为形式推导，不过有的是直接根据(∈)，$(\to_-)$判定的，而有的则是根据(τ)，$(\to_+)$从前边已有的形式推导判定的。最后判定的则正是所要的形式推导。所以可以说，这个过程是判定$A\to(B\to C),A\to B\vdash A\to C$的形式证明。在不致混淆时，常将上述的"形式"均省略掉。

对上述例9所给的形式证明，可以一种更简单明了的斜形证明表示之，为表明相应关系，斜形证明的步号与上述的相一致。如下：

(1) $A\to(B\to C)$ （原来的第一个前提）

(2) $A\to B$ （原来的第二个前提）

(3) A （新加一个前提）

(4) $B\to C$ (3)(1)$(\to_-)$

(6) B (3)(2)$(\to_-)$

(8) C (6)(4)$(\to_-)$

(10)　$A\to C$　　(3)(8)($\to_+$)

可以看到，斜形证明所用的前提是其上方和左上方的，比如说，(8)是利用了(4)、(6)根据($\to_-$)而得，(6)、(4)的公式均在(8)的上方；而(6)是利用了(3)、(2)根据($\to_-$)得到，(3)在(6)的上方，(2)在(6)的左上方；(10)尽管是从(3)的公式 A 所得(8)的公式 C 而来，但其并未使用(3)作为前提，这点儿要特别注意。另外，斜形证明较推导序列给出的步数要少，且可以看出，省掉了(τ)规则的运用，且($\in$)规则的运用也省掉了(实际上是换成了给出前提)。所以能这样作，主要是：凡处在某一公式上方和左上方的公式均可作为这公式的前提使用，而处在下方和右下方的公式又都是其上方和左上方的结论，不用使用($\in$)和(τ)，自然而然是可以作为前提使用，或作为结论而导出。

例10—33　试给出下述推导的证明。

【10】$A\vdash A$　(称作同一律)；

【11】$A\vdash B\to A$(肯定后件律，而($\in$)也可称作肯定前件律)；

【12】$A\to B, B\to C\vdash A\to C$(蕴涵传递律)；

【13】$A\to(B\to C), A\to B\vdash A\to C$(蕴涵分配律)；(亦即，$A\to(B\to C)\vdash(A\to B)\to(A\to C)$)

【14】$\neg A\vdash A\to B$(否定前件律)；

【15】如果 $\Gamma, A\vdash B, \neg B$，则 $\Gamma\vdash\neg A$(归谬律)；

【16】$A\to B\vdash\neg B\to\neg A$(换位律)；

【17】$A\wedge B\dashv\vdash B\wedge A$(合取交换律)；

【18】$(A\wedge B)\wedge C\dashv\vdash A\wedge(B\wedge C)$(合取结合律)；

【19】$\vdash\neg(A\wedge\neg A)$(无矛盾律)；

【20】$\vdash A\vee\neg A$(排中律)；

【21】$\neg(A\wedge B)\dashv\vdash\neg A\vee\neg B$，　$\neg(A\vee B)\dashv\vdash\neg A\wedge\neg B$(德·摩尔根律)

【22】$A\to B\dashv\vdash\neg A\vee B$；

【23】$A\leftrightarrow B\dashv\vdash A\to B, B\to A$；

【24】$\forall x\forall yA(x,y)\dashv\vdash\forall y\forall xA(x,y)$；

【25】$\exists x\exists yA(x,y)\dashv\vdash\exists y\exists xA(x,y)$；

【26】$\exists x\forall yA(x,y)\vdash\forall y\exists xA(x,y)$；

【27】$\forall xA(x)\dashv\vdash\neg\exists x\neg A(x)$；

【28】$\exists xA(x)\dashv\vdash\neg\forall x\neg A(x)$；

【29】$\forall x(A(x)\rightarrow B(x)),\forall xA(x)\vdash\forall xB(x)$；

【30】$A\rightarrow\forall xB(x)\dashv\vdash\forall x[A\rightarrow B(x)]$，$x$ 不在 A 中出现；

【31】$\exists xA(x)\rightarrow B\dashv\vdash\forall x[A(x)\rightarrow B]$，$x$ 不在 B 中出现；

【32】$\forall xA(x)\wedge\forall xB(x)\dashv\vdash\forall x[A(x)\wedge B(x)]$；

【33】$\exists x[A(x)\wedge B(x)]\vdash\exists xA(x)\wedge\exists xB(x)$。

例34　试给出推导

$\exists y\forall x(A(x)\leftrightarrow x\equiv y)\dashv\vdash\underbrace{\exists x(A(x)\wedge\forall y(A(y)\rightarrow y\equiv x))}_{\text{常简记为 }\exists!xA(x)}$的形式证明。

下面，以斜形证明给出之：

(1) $\exists y\forall x(A(x)\leftrightarrow x\equiv y)$，

(2) 　　$\forall x(A(x)\leftrightarrow x\equiv b)$，$b$ 不在 $A(x)$ 中出现

(3) 　　$A(b)\leftrightarrow b\equiv b$ 　　(2)($\forall_-$)

(4) 　　$b\equiv b$ 　　(I_+)

(5) 　　$A(b)$ 　　(3)(4)($\leftrightarrow_-$)

(6) 　　　　$A(a)$，a 不同于 b，且不在 $A(x)$ 中出现

(7) 　　　　$A(a)\leftrightarrow a\equiv b$ 　　(2)($\forall_-$)

(8) 　　　　$a\equiv b$ 　　(6)(7)($\leftrightarrow_-$)

(9) 　　$A(a)\rightarrow a\equiv b$ 　　(6)(8)($\rightarrow_+$)

(10) 　　$\forall y(A(y)\rightarrow y\equiv b)$ 　　(9)($\forall_+$)

(11) 　　$A(b)\wedge\forall y(A(y)\rightarrow y\equiv b)$ 　　(5)(10)($\wedge_+$)

(12) 　　$\exists x(A(x)\wedge\forall y(A(y)\rightarrow y\equiv x))$ 　　(11)($\exists_+$)

(13) $\exists x(A(x)\wedge\forall y(A(y)\rightarrow y\equiv x))$ 　　(2)(12)($\exists_-$)

故有 $\exists y\forall x(A(x)\leftrightarrow x\equiv y)\vdash\exists x(A(x)\wedge\forall y(A(y)\rightarrow y\equiv x))$；

(1) $\exists x(A(x)\wedge\forall y(A(y)\rightarrow y\equiv x))$

(2) 　　$A(a)\wedge\forall y(A(y)\rightarrow y\equiv a)$，$a$ 不在 $A(x)$ 中出现

(3) $A(b)$，b 不在 $A(x)$ 中出现，且 $b \neq a$

(4) $A(b) \to b \equiv a$ (2)($\wedge_-$)($\forall_-$)

(5) $b \equiv a$ (3)(4)($\to_-$)

(6) $b \equiv a$

(7) $A(a)$ (2)($\wedge_-$)

(8) $A(b)$ (6)(7)(I_-)

(9) $A(b) \leftrightarrow b \equiv a$ (3)(5)(6)(8)($\leftrightarrow_+$)

(10) $\forall x(A(x) \leftrightarrow x \equiv a)$ (9)($\forall_+$)

(11) $\exists x \forall x(A(x) \leftrightarrow x \equiv y)$ (10)($\exists_+$)

(12) $\exists y \forall x(A(x) \leftrightarrow x \equiv y)$ (2)(11)($\exists_-$)

故有 $\exists y \forall x(A(x) \leftrightarrow x \equiv y) \vdash\dashv \exists x(A(x) \wedge \forall y(A(y) \to y \equiv x))$。

读者会注意到，在2.2中讨论后承关系时，是说公式集合 Φ 和一个公式 φ 的关系，即 $\Phi \models \varphi$；而在本节中讨论形式推导关系时，则是说的公式的有穷序列 Γ 和一个公式 A 的关系。事实上，也可讨论公式集合 Φ 和一个公式 φ 间的形式推导关系。有如下两种方法：

(1)直接给出定义：称**公式集合 Φ 和公式 φ 满足形式推导关系**，如果有 $\Gamma \subseteq \Phi$，使 $\Gamma \vdash \varphi$；此时，也记作 $\Phi \vdash \varphi$；

(2)**修改形式推理规则，使成为公式集合 Φ 和公式 φ 之间的规则**，比如将($\in$)改为：

[$\in$] $\Phi \vdash \varphi, \varphi \in \Phi$；

将(τ)改为：

[τ] 如果 $\Phi \vdash A_1 \cdots, A_n \vdash \varphi$，则 $\Phi \vdash \varphi$； 而将($\to_-$)改为：

[$\to_-$] 如果 $A, A \to B \in \Phi$，则 $\Phi \vdash B$； 将($\to_+$)改为：

[$\to_+$] 如果 $\Phi \cup \{A\} \vdash B$，则 $\Phi \vdash A \to B$； 将($\neg$)改为：

[$\neg$] 如果 $\Phi \cup \{\neg A\} \vdash B, \neg B$，则 $\Phi \vdash A$；其他各条均类似地修改。那么有：

$\Phi \vdash \varphi$ 当且仅当有 $\Gamma \subseteq \Phi, \Phi \vdash \varphi$。

由此可见，这两种改法是等价的。只不过第(1)种改法是将不能行的过程推到"后来"，而第(2)种改法，则在"一开始"即意味着不能

行了。

所以从此后，即可讨论 $\Phi\vdash\varphi$ 这样的形式推导了。

2.4 形式推导的正确性和协调性

形式推导 $\Phi\vdash\varphi$ **是正确的**（或可靠的），如果有 $\Phi\models\varphi$。

首先，对推理规则中的任何"$\vdash$"，均改为"$\models$"，则每条规则均是成立的。这里，举几例证明之：

$(\in)A_1,\cdots,A_n\models A_i\qquad(i=1,\cdots,n)$；显然成立；

(τ)若 $\Gamma\models\triangle$，且 $\triangle\models A$，则 $\Gamma\models A$：

可设 $\triangle=A_1,\cdots,A_n$，那么 $\Gamma\models\triangle$，即 $\Gamma\models A_1,\cdots,\Gamma\models A_n$；这就是说，对 Γ 的任何模型 $\mathfrak{I}$，也均为 $A_i(i=1,\cdots,n)$ 的模型，而从 $\triangle\models A$，故也为 A 的模型，证毕；

$(\exists_+)A(t)\models\exists xA(x)$，$t$ 为项，而 $A(x)$ 为将 $A(t)$ 中某些 t 换成 x 而得：

对 $A(t)$ 的任何模型 $\mathfrak{I}$，即 $\mathfrak{I}\models A(t)$，可设 $\mathfrak{I}(t)=a\in A$，那么，解释 $\mathfrak{I}\frac{a}{x}$ 则是 $A(x)$ 的模型（$\because\mathfrak{I}\frac{a}{x}$ 将 x 赋值为 a，而其他均同 $\mathfrak{I}$，故在 $\mathfrak{I}\frac{a}{x}$ 下，$A(x)$ 成立当且仅当在 $\mathfrak{I}$ 下，$A(t)$ 成立，故由语义定义，$\mathfrak{I}\models\exists xA(x)$。

$(\exists_-)$若 $A(a)\models B$，且 B 中无 a 出现，则有 $\exists xA(x)\models B$：

设任 $\mathfrak{I}\models\exists xA(x)$，现证 $\mathfrak{I}\models B$：从 $\mathfrak{I}\models\exists xA(x)$，由语义定义，必有 $b\in A$，使 $\mathfrak{I}\frac{b}{x}\models A(x)$，那自然也有 $\mathfrak{I}\frac{b}{a}\models A(a)$，

故由 $A(a)\models B$，必有 $\mathfrak{I}\frac{b}{a}\models B$，但 B 中无 a 出现，故 $\mathfrak{I}$，$\mathfrak{I}\frac{b}{a}$ 对 B 的解释相同，因而 $\mathfrak{I}\models B$，证毕。

其他各条规则均有类似情况，读者试补证之。

定理4a 对任 Φ 和 φ，$\Phi\vdash\varphi$ 均是正确的。

证明由直接定义，可设有 $\Gamma\subseteq\Phi$，使 $\Gamma\vdash\varphi$，那么，必有 $\Gamma\vdash\varphi$ 的一个形式证明

$\Gamma_1\vdash A_1,\cdots,\Gamma_n\vdash A_n$；且 $\Gamma_n=\Gamma,A_n=\varphi$，

故施纳于长度 n，利用推理规则的正确性，立证 $\Gamma \models A_n$，故 $\Phi \models \varphi$。证毕。

称 Φ 是**协调的**，如果没有公式 φ，使

$$\Phi \vdash \varphi \quad 且\ \Phi \vdash \neg \varphi。$$

定理5 若 Φ 可满足，则 Φ 协调。

证明 Φ 可满足，即有 $\mathfrak{I} \models \Phi$；若 Φ 不协调，则有 φ，使 $\Phi \vdash \varphi$，$\neg \varphi$，故由**定理4a**，立得 $\Phi \models \varphi, \neg \varphi$，那么，对上述 $\mathfrak{I}$，也就有

$\mathfrak{I} \models \varphi, \neg \varphi$

这是不可能的，故证毕。

事实上，由**定理5**也可证得**定理4a**，即二者是相等价的。留作练习。

由 $\Phi \vdash \varphi$ 当且仅当有 Φ_0（有穷），$\Phi_0 \subseteq \Phi$，使 $\Phi_0 \vdash \varphi$，故从协调的定义，立得：

定理6 Φ 协调当且仅当对任意 Φ_0（有穷），$\Phi_0 \subseteq \Phi$，Φ_0 协调。

定理7 对每个 $n \in N$，设 S_n 是符号集，且有 $S_0 \subset S_1 \subset S_2 \subset \cdots$，而 Φ_n 是 S_n-公式集，Φ_n 均相对于 S_n 协调（即没有 S_n-公式 φ，使 $\Phi_n \vdash \varphi, \neg \varphi$），$\Phi_0 \subset \Phi_1 \subset \Phi_2 \subset \cdots$。令 $S = \bigcup_{n \in N} S_n$，$\Phi = \bigcup_{n \in N} \Phi_n$。则有：$\Phi$ 协调（自然是相对于 S）。

证明 设 Φ 相对于 S 不协调，则由**定理6**必有有穷集 $\Psi \subseteq \Phi$，使 Ψ 不协调。由 Ψ 有穷，故可设有 k，使 $\Psi \subseteq \Phi_k$。因而 Φ_k 也不协调，不妨设 $\Phi_k \vdash a \equiv a, \neg a \equiv a$。那么，必有有穷的 $\Gamma_k \subseteq \Phi_k$，$\Gamma_k \vdash, a \equiv a, \neg a \equiv a$，且可设两个推导的证明均已构造出，那么，其中只有有穷多个符号，故其中所出现的公式均属于某 L^{S_m}。不妨设 $m \geqslant k$，那么，两个推导的证明均是 L^{S_m} 作为语言的形式证明，因此 Φ_k 相对于 S_m 不协调。再由 $\Phi_k \subseteq \Phi_m$，故 Φ_m 相对于 S_m 也不协调，这与假设相矛盾。故定理7证毕。

2.5 不含量词公式的合取（析取）范式和含量词公式的前束范式

称一个公式为**合取（析取）范式**，如果其中不含量词，且其形

如：

$$(A_{11}\vee A_{12}\vee\cdots\vee A_{1m_1})\wedge(A_{21}\vee A_{22}\vee\cdots\vee A_{2m_2})$$

$\wedge\cdots\wedge(A_{n_1}\vee\cdots\vee A_{nm_n})$，其中，每个 A_{ij} 中最多只含逻辑连接词“$\neg$”(上述式中“$\wedge$”与“$\vee$”相交换，余不变)。

那么，有：

定理 8 任一公式 A(不含量词)，均有与其等值的公式 A'(即 $A\dashv\vdash A'$)，且 A' 为合取(析取)范式。

证明 **试练习证明之。**

称公式 A 是**前束范式**，如果其形如：

$Q_1x_1Q_2x_2\cdots Q_nx_nB$，

其中，Q_i 为$\exists$ 或$\forall$ ，而 B 中无量词。那么有：

定理 9 任公式 A，均有与其相等值的公式 A'，且 A' 为前束范式。

证明 **试练习证明之。**

由定理8和9，只要将前束范式中的 B(称作**母式**，而 $Q_1X_1\cdots Q_nX_n$ **称作前束词**)再化成合取或析取范式，则所得的公式很规范，对处理不少问题很有帮助。这里不多介绍了。

2.6 形式推导的完全性

对任给的公式集 Φ 和公式 φ，如果从 $\Phi\models\varphi$，能证得也有 $\Phi\vdash\varphi$，则称该**一阶逻辑系统的形式推导是完全的**，或者称所给的推理规则是充分的。

正像**定理** 4和**定理** 5相等价，非常易得：形式推导的完全性等价于：**若 Φ 协调，则 Φ 可满足**。本节，即通过证这等价形式而证明完全性。不过，为节省篇幅，这里仅对 S 是可数的情况加以证明；且去掉“$\rightarrow$”、“$\leftrightarrow$”及相应的规则。

首先，给 S 中加入可数个新的个体词 c_i $(i\in N)$。令 $C=\{c_i|i\in N\}$，将 S 修改为 $S'=S\cup C$。

对任给 $\varphi\in L^S$，定义和其相对的公式 $\varphi\neg$：

(1) 若 φ 为原子公式，则 $\varphi\neg$ 即$\neg\varphi$；

(2)$(\neg\varphi)^{\neg}$即 φ;

(3)$(\varphi_1\wedge\varphi_2)^{\neg}$即$\neg\varphi_1\vee\neg\varphi_2$;

(4)$(\varphi_1\vee\varphi_2)^{\neg}$即$\neg\varphi_1\wedge\neg\varphi_2$;

(5)$(\forall x\varphi(x))^{\neg}$即 $\exists x\neg\varphi(x)$;

(6)$(\exists x\varphi(x))^{\neg}$即 $\forall x\neg\varphi(x)$。

令$\mathbb{S}\subseteq\{\mathscr{S}\mid\mathscr{S}\subseteq L^S,\mathscr{S}$ 可数$\}$。称$\mathbb{S}$ **是协调性质**，如果任$\mathscr{S}\in\mathbb{S}$，均满足：

(C_1)(协调规则)或者 $\varphi\notin\mathscr{S}$,或者$\neg\varphi\notin\mathscr{S}$;

(C_2)($\neg$-规则)如果$(\neg\varphi)\in\mathscr{S}$,则 $\mathscr{S}\cup\{\varphi^{\neg}\}\in\mathbb{S}$;

(C_3)($\wedge$-规则)如果$(\varphi_1\wedge\varphi_2)\in\mathscr{S}$,则对 $i=1,2$,均有$\mathscr{S}\cup\{\varphi_i\}\in\mathbb{S}$;

(C_4)($\forall$-规则)如果$(\forall x\varphi(x))\in\mathscr{S}$,则对任 $c_i\in C$,$\mathscr{S}\cup\{\varphi(c_i)\}\in\mathbb{S}$;

(C_5)($\vee$-规则)如果$(\varphi_1\vee\varphi_2)\in\mathscr{S}$,则有 $\varphi=\varphi_1$或 φ_2,$\mathscr{S}\cup\{\varphi\}\in\mathbb{S}$;

(C_6)($\exists$-规则)如果$(\exists x\varphi(x))\in\mathscr{S}$,则对某 $c_i\in C$,$\mathscr{S}\cup\{\varphi(c_i)\}\in\mathbb{S}$;

(C_7)(等词-规则)设 t 是**基本项**,$c,d,\in C$,则：

如果$(c\equiv d)\in\mathscr{S}$,则 $\mathscr{S}\cup\{d\equiv c\}\in\mathbb{S}$,

如果 $c\equiv t,\varphi(t)\in\mathscr{S}$,则 $\mathscr{S}\cup\{\varphi(c)\}\in\mathbb{S}$,

且对某 $e\in C$,$\mathscr{S}\cup\{e\equiv t\}\in\mathbb{S}$;

这里,**基本项**包括所有的**个体词**和 $fc_{i_0}c_{i_1}\cdots c_{i_{n-1}}$,$f$ 是函数词,$c_{i_j}\in C$ $(j=0,1,\cdots,n-1)$。

那么,易证：

定理10 若$\mathbb{S}$ 是协调性质,则：

(1)$\mathbb{S}'=\{\mathscr{S}'\mid\mathscr{S}'\subseteq\mathscr{S},\mathscr{S}\in\mathbb{S}\}$是协调性质;

(2)$\mathbb{S}''=\{\mathscr{S}'\mid\mathscr{S}'\supseteq\mathscr{S},\mathscr{S}'$有穷$,\mathscr{S}\in\mathbb{S}\}$是协调性质。

证明 略。读者试证明之。

定理11 设$\mathbb{S}$ 是协调性质,且 $\mathscr{S}_0\in\mathbb{S}$,那么:$\mathscr{S}_0$有模型。

证明 利用**定理10**,可设$\mathbb{S}$ 满足:若$\mathscr{S}\in\mathbb{S}$,$\mathscr{S}'\subseteq\mathscr{S}$,则 $\mathscr{S}'\in$

$\mathbb{S}$。设 $X=\{\varphi_0,\varphi_1,\varphi_2,\cdots\}$ 是 L^S 的所有公式集，$T=\{t_0,t_1,t_2,\cdots\}$ 是所有的基本项集。如下选择公式集的序列 $\mathscr{S}_0,\mathscr{S}_1,\cdots$，使满足：对 $n+1\geqslant 1$，$j\leqslant n$

(1) $\mathscr{S}_n\subset\mathscr{S}_{n+1}\in\mathbb{S}$；

(2) 如果 $\mathscr{S}_n\cup\{\varphi_j\}\in\mathbb{S}$，则 $\varphi_n\in\mathscr{S}_{n+1}$；

(3) 如果 $\mathscr{S}_n\cup\{\varphi_j\}\in\mathbb{S}$，且 $\varphi_n=A\vee B$，则或者有 $A\in\mathscr{S}_{n+1}$，或者有 $B\in\mathscr{S}_{n+1}$；

(4) 如果 $\mathscr{S}_n\cup\{\varphi_j\}\in\mathbb{S}$，且 $\varphi_n=\exists x\varphi(x)$，则对某 $c\in C$，有 $\varphi(c)\in\mathscr{S}_{n+1}$；

(5) 有 $c\in C$，使 $(c\equiv t_n)\in\mathscr{S}_{n+1}$。

由协调性质的定义，知可以选到上述公式集的序列。

令 $\mathscr{S}_\omega=\bigcup_{n<\omega}\mathscr{S}_n$。定义 C 上的关系"$\sim$"：

任 c，$d\in C$，$c\sim d$ 当且仅当 $(c\equiv d)\in\mathscr{S}_\omega$。

那么，从 (C_7) 和上述序列的选择，易知："$\sim$"是等价关系。兹仅例证一下传递性：设 c，d，$e\in C$，且 $c\sim d$，$d\sim e$，现证 $c\sim e$。由 $c\sim d$，$d\sim e$，可设有 n_1，n_2，使 $(c\equiv d)$，$(d\equiv e)$ 分别属于 $\mathscr{S}_{n_1}$，$\mathscr{S}_{n_2}$；因为 X 中有所有的 L^S 中的公式，故可设 $(c\equiv e)$ 为某 φ_m；由 $\mathscr{S}_0\subset\mathscr{S}_1\subset\mathscr{S}_2\subset\cdots$，所以可设有 $n\geqslant m$，使得：

$$(c\equiv d),(d\equiv e)\in\mathscr{S}_n,\cdots\cdots(*)$$

利用 $\mathbb{S}$ 为协调性质的 (C_7)：如果 $(c\equiv t),\varphi(t)\in\mathscr{S}$，则 $\mathscr{S}\cup\{\varphi(c)\}\in\mathbb{S}$；故从 $(*)$，取 $t=d$，$\varphi(t)=(d\equiv e)$，$\mathscr{S}=\mathscr{S}_n$，立得 $\mathscr{S}_n\cup\{(c\equiv e)\}\in\mathbb{S}$，结合 $\mathscr{S}_m\subset\mathscr{S}_n$，故 $\mathscr{S}_m\cup\{(c\equiv e)\}$ 也属于 $\mathbb{S}$，因而 $(c\equiv e)\in\mathscr{S}_{m+1}$，从而 $c\sim e$。

取 $A=\{\{c\}\mid c\in C$，$\{c\}$ 为"$\sim$"的等价类$\}$。而由 (C_7)，如果 $\varphi(c_1,\cdots,c_n)\in\mathscr{S}_\omega$，$c_1\sim d_1$，$\cdots c_n\sim d_n$，那么，$\varphi(d_1,\cdots,d_n)\in\mathscr{S}_\omega$. 所以，可取 S'-结构 $\mathfrak{A}=(A,\alpha)$，且 α 将个体词均对应为相应的等价类，而将函数词 f 均对应它们自己，从基本项的定义和 (C_7)，α 即将相应的基本项对应为相应的等价类，α 将谓词也对应它们自己。

任意取 $\mathfrak{A}$ 中的赋值 β。那么，S' 解释 $\mathfrak{I}=(\mathfrak{A},\beta)$ 对原子公式如

下解释：

(6) 如果 t 是基本项，$c\in C$，则

$\mathfrak{I}\models(c\equiv t)$ 当且仅当 $(\{c\}\equiv\mathfrak{I}(t))\in\mathscr{S}_\omega$，

(7)如果 R 是 n-元谓词符号，$c_{i_0},\cdots,c_{i_{n-1}}\in C$，

则 $\mathfrak{I}\models Rc_{i_0}\cdots c_{i_{n-1}}$ 当且仅当 $Rc_{i_0}\cdots c_{i_{n-1}}\in\mathscr{S}_\omega$。

易知，(6) 和 (7) 即完全确定了 $\mathfrak{I}$ 对任何 $L^{S'}$ 公式的解释。事实上，施归纳于项的定义，由(C_1)、(C_7)即知对 $\mathscr{S}_\omega$中的任原子公式或其否定，赋予了真值；而由(C_2)—(C_6)，即知：对 $\mathscr{S}_\omega$中的任一公式都赋予了真值。因而，$\mathfrak{I}$ 也是 $\mathscr{S}_0$的模型。**定理11**证毕。

定理12（完全性定理） Φ 协调，则 Φ 可满足。

证明 令 $\mathbb{S}=\{\mathscr{S}\mid\mathscr{S}\subseteq L^{S'},\mathscr{S}$ 协调，且 $\mathscr{S}$ 中只含有穷个新常量$\}$，那么，只要证明$\mathbb{S}$ 是协调性质，即证毕。因为由**定理11**，任$\mathscr{S}\in\mathbb{S}$，$\mathscr{S}$ 均有模型。而又不难知道，一定有一$\mathscr{S}$，使 $\Phi\subseteq\mathscr{S}\in\mathbb{S}$，故 Φ 也有模型。

为证$\mathbb{S}$ 是协调性质，必须按定义，逐条验证(C_1)—(C_7)。这里仅以验证(C_5)和(C_6)为例：

(C_5)($\vee$-规则)，如果$(\varphi_1\vee\varphi_2)\in\mathscr{S}$，

但 $\mathscr{S}\cup\{\varphi_1\}$，$\mathscr{S}\cup\{\varphi_2\}$均不属于$\mathbb{S}$，

那么，由$\mathbb{S}$ 的定义，则 $\mathscr{S}\cup\{\varphi_1\}$，$\mathscr{S}\cup\{\varphi_2\}$均不协调，再由**定理6**，必有 $\mathscr{S}\cup\{\varphi_i\}$ 的有穷子集，均不协调。那么必得：$\mathscr{S}\cup\{\varphi_1\vee\varphi_2\}=\mathscr{S}$ 不协调，矛盾。故 (C_5) 验证毕。

(C_6)($\exists$-规则) 如果$\exists x\varphi(x)\in\mathscr{S}$，且对任 $c_i\in C$，$\mathscr{S}\cup\{\varphi(c_i)\}$ 均不属于$\mathbb{S}$，

那么，$\mathscr{S}\cup\{\varphi(c_i)\}$均不协调，不难得到：

$\mathscr{S}\cup\{\exists x\varphi(x)\}=\mathscr{S}$ 也不协调。矛盾。验证毕。

完全性定理，是一阶逻辑的一个最基本的定理。证明起来，也有相当的难度。自从 Gödel 在1930年证明后（那是Gödel 的博士论文），又有 Henkin 1949年的证明，还有1956年的一个拓扑证明方法。上述所给的证明，取自 Keisler 证明无穷逻辑 $L_{\omega_1\omega}$（后面还要作介绍）的完全性的方法。建议有兴趣的读者去参考之。

2.7 一阶逻辑的局限性、Gödel 不完全定理

一阶逻辑很简洁，能表达很多数学系统；且其推理系统又具有正确性和完全性。按说，这些优点应很让人鼓舞。但随着 Gödel 不完全性定理的证明，一阶逻辑的局限性便越来越显现出来。本节，即集中地讨论这些局限性。

定理13 设$|A|$表示集合 A 的势，则：一阶语言的势$|L^S|\leqslant|S|+\aleph_0$。

证明 略.

定理14 (Löwenheim-Skolem)

(1) 设$|L^S|=\aleph_0$,且 $\Phi\subseteq L^S$可满足，则 Φ 在可数模型中可满足；

(2)设$|L^S|\geqslant\aleph_0$,且$\Phi\subseteq L^S$可满足，则 Φ 在$\leqslant|L^S|$的模型中可满足。

证明 (1) 从 Φ 可满足，由**定理5**（即正确性定理的另种表述），Φ 协调；那么，类似完全性定理的证明，可得 Φ 的模型，而其论域 A 可数。故证毕；

(2) 这个证明要借助于对$|S|$为各种基数时的完全性定理的证明。因没有介绍，故这里仅作一概略证明（或称说明）。类似(1) 的证明，由 Φ 可满足，故 Φ 协调，在构造 Φ 的模型时，其论域 A 由三部分构成：一是原有的个体词数，其势$\leqslant|L^S|$，二是具有形如$\exists x\varphi(x)$的公式之数$\leqslant|L^S|$，三是新增加的个体词，其势$\leqslant|L^S|$，故 A 之势$|A|\leqslant|L^S|$。说明完毕。

定理15 (Lowenheim-Skolem-Tarski) 设 $\Phi\subseteq L^S$在无穷模型中可满足，则对任何势 $\lambda\geqslant|L^S|$，均有势为 λ 的模型，Φ 在其中可满足。

证明 设 Φ 在无穷模型 $\mathfrak{I}$ 中可满足（即其结构 $\mathfrak{A}=(A,\alpha)$ 的论域 A 无穷）。

令 $S'=S\cup C$，而 C 是 λ 个不同的个体词集合。构造 $L^{S'}$的公式集合 $\Sigma=\{\neg c\equiv c'|c,c'\in C\}$。那么，$\Sigma\cup\Phi$ 的每个有穷子集在 $\mathfrak{I}$ 中

均可满足，这只要将其中出现的有穷多个新的不同的个体词（即在有穷子集中出现的个体词）对应到 A 的不同对象即可（由于 A 无穷，故这总可作到）。因而，由紧致性定理（由**完全性定理**和**定理6**、可得紧致性定理的另种形式：Φ 可满足当且仅当 Φ 的任有穷子集 Φ_0 可满足），$\Sigma\cup\Phi$ 也是可满足的。由**定理14**，$\Sigma\cup\Phi$ 在某势$\leqslant\lambda$ 的模型中可满足（$\because |L^{S'}|=|L^S|+\lambda=\lambda$）。但易知，$\Sigma\cup\Phi$ 的任何模型之势必$\geqslant\lambda$，故得：$\Sigma\cup\Phi$ 在势为 λ 的模型中可满足，因而，Φ 也在这模型中可满足。定理证毕。

系16 若$|L^S|=\aleph_0$，且 $\Phi\subseteq L^S$有无穷模型，则 Φ 有各种不同基数的模型。

正是有了一阶逻辑的完全性、紧性和模型基数的不确定性定理，才引出了一阶逻辑模型论的丰富内容。但是，紧性和模型基数的不确定性，也正说明了一阶逻辑之不足。即紧性说明了推理的递归性，而基数的不确定性，说明了一阶逻辑刻画不清楚各种不同的基数。这些都是与非抽象的数学系统（即只有一个模型的数学系统）不一致的。

下面定理说明：一阶逻辑无法刻画任意的有限模型。

定理17 若 $\Phi\subseteq L^S$有任意的有限模型，则 Φ 必有无限模型。

证明 令 $\Sigma=\Phi\cup\{\varphi_{\geqslant n}|n\geqslant 2\}$。那么，可知：$\Sigma$ 的任有穷集 Σ_0 均会有模型（事实上，均为有限模型），故由紧性定理，Σ 也有模型。不难知道：Σ 的任何模型必为无穷模型。证毕。

由该定理，有限群、有限域以及任何有限结构的类，一阶逻辑均无法刻画。

定理18 挠群（Torsion）不能由一阶句子集刻划。（挠群，即满足 $\forall x\exists n\geqslant 1[\underbrace{x\circ\cdots\circ x}_{n\text{次}}=e]$的 Abel 群）

证明 设有 Φ，其刻画了挠群。

令 $\Sigma=\Phi\cup\{\neg\underbrace{c\circ\cdots\circ c}_{n\text{次}}\equiv e|n\geqslant 1\}$，

那么，易知：任有穷集 $\Sigma_0\subset\Sigma$ 均有模型。（事实上，可设 $\Sigma_0\subset\Phi\cup$

$\{\neg \underbrace{co\cdots oc}_{n次}=e \mid 1<n<n_0\}$；那么，每个阶为 n_0 的循环群必为 Σ_0 的模型，这只要将 c 对应为生成元即可）故由紧性定理，Σ 有模型，设为 $\mathfrak{I}=(\mathfrak{A},\beta)$，那么，$c^{\mathfrak{A}}$ 有无穷阶，但该 $\mathfrak{I}$ 又是 Φ 的模型，这便与挠群的元均为有穷阶相矛盾。证毕。

在 2.2 的例 8 中，给出了自然数的算术的公理，而且解释了其中的归纳公理，当其为二阶公式时，即刻画了自然数（从范畴的意义上讲，即在同构意义下，自然数集合是唯一的），但当把该公理换成可数无穷条一阶公式时，则出了问题，就是说这可数无穷条一阶公式也抵不上那一条二阶公式。现在再进一步研究这一问题。这即是 **Gödel 不完全性定理**：设 Φ_{ar}^{I} 为自然数的算术的一阶公理集，且设 Φ_{ar}^{I} 是 ω-**协调的**，即对任公式 $A(a)$，均不会同时有

$$\Phi_{ar}^{I}\vdash A(\underline{0}),A(\underline{1}),A(\underline{2}),\cdots;\Phi_{ar}^{I}\vdash \neg\forall xA(x)$$

均成立，则总有一公式 $A\in L^{S_{ar}}$，$S_{ar}=\{+,\cdot,\underline{0},\underline{1}\}$，使 $\Phi_{ar}^{I}\nvdash A$，$\neg A$。

其中，$\underline{2}=\underline{1}+\underline{1}$，$\underline{3}=\underline{2}+\underline{1}$，$\cdots$。

这里，不打算给出该定理的全部证明，而仅揭示出其中最本质的东西。

称 N 上的关系 R（n-元的）在 Φ_{ar}^{I} 中可表示，如果有公式 $\varphi(a_0,\cdots,a_{n-1})\in L^{S_{ar}}$，使对任给的

$$(m_0,\cdots,m_{n-1})\in N^n,$$

均有：$R(m_0,\cdots,m_{n-1})\Rightarrow\Phi_{ar}^{I}\vdash\varphi(\underline{m_0},\cdots,\underline{m_{n-1}})$，

$$\tilde{R}(m_0,\cdots,m_{n-1})\Rightarrow\Phi_{ar}^{I}\vdash\neg\varphi(\underline{m_0},\cdots,\underline{m_{n-1}}),$$

并称 φ 表示了 R。

定理19 设 Φ_{ar}^{I} 是 ω-协调的，且 N 上的二元关系 $\mathscr{A}(x,y)$ 在 Φ_{ar}^{I} 中由公式 $A(a,b)$ 表示，并有集合

$$Q=\{x\mid 有\ y\ 使\ \mathscr{A}(x,y)\}$$

是递归可枚举而非递归的。则 Φ_{ar}^{I} 不完全，即有 $A\in L^{S_{ar}}$，使 $\Phi_{ar}^{I}\nvdash A$，$\neg A$。

证明 令 $Q'=\{x\mid\Phi_{ar}^{I}\vdash\exists yA(\underline{x},y)\}$，那么易知：$Q'=Q$。

事实上，设 $x\in Q$，即有 $y\in N$，使 $\mathscr{A}(x,y)$，那么由 $\mathscr{A}(x,y)$ 在 Φ_{ar}^{I} 中由 $A(a,b)$ 表示，故有

$$\Phi_{ar}^{I}\vdash A(\underline{x},\underline{y}),$$

使用 $(\exists_{+})$ 得：$\Phi_{ar}^{I}\vdash\exists yA(\underline{x},y)$，故 $x\in Q'$；

反过来，设 $x\in Q'$，则有：

$$\Phi_{ar}^{I}\vdash\exists yA(\underline{x},y),$$

易得　$\Phi_{ar}^{I}\vdash\neg\forall y\neg A(\underline{x},y)$；而若 $x\notin Q$，则有

$$\widetilde{\mathscr{A}}(x,0),\widetilde{\mathscr{A}}(x,1),\widetilde{\mathscr{A}}(x,2),\cdots,$$

故

$$\Phi_{ar}^{I}\vdash\ \neg A(\underline{x},\underline{0}),\neg A(\underline{x},\underline{1}),\neg A(\underline{x},\underline{2}),\cdots,$$

这与 Φ_{ar}^{I} 是 ω-协调的相矛盾，故 $x\in Q$；

这样，$Q'=Q$ 证毕。

令 $R=\{x\,|\,\Phi_{ar}^{I}\vdash\neg\exists yA(\underline{x},y)\}$，

那么，易知：$R\subset\overline{Q'}$。事实上，设 $x\in R$，故 $\Phi_{ar}^{I}\vdash\neg\exists yA(\underline{x},y)$，由 Φ_{ar}^{I} 是 ω-协调的，故 Φ_{ar}^{I} 也协调，因而，$\Phi_{ar}^{I}\nvdash\exists yA(\underline{x},y)$，所以 $x\in\overline{Q'}$；另外易知：R 是递归可枚举的，但是 $\overline{Q'}$ 却是非递归可枚举的（因为，否则，$Q=Q'$ 即为递归的了，与定理的条件矛盾）。因而，$R\subsetneq\overline{Q'}$。这就是说，有 $n\in\overline{Q'}-R$，即该 n 使

$$\Phi_{ar}^{I}\nvdash\exists yA(\underline{n},y),$$

且 $\Phi_{ar}^{I}\nvdash\neg\exists yA(\underline{n},y)$。定理证毕。

定理 19 和 Gödel 不完全性定理有二个区别：

(1) **定理 19** 中的公式 $\exists yA(\underline{n},y)$（这种公式称作是 Φ_{ar}^{I} 不可判定的）是存在性的，而在 Gödel 定理中却是直接构造出来的。不过，从这定理的证明中，还可见：$\overline{Q'}-R$ 也是非递归可枚举的（否则，$(\overline{Q'}-R)\cup R=\overline{Q'}$ 也成为递归可枚举的了），故有非递归可枚举个 $n\in\overline{Q'}-R$，使相应的公式 $\exists yA(\underline{n},y)$ 是 Φ_{ar}^{I} 不可判定的。

(2) 定理19中比 Gödel 定理中增加了个条件："N 上的二元关系 $\mathscr{A}(x,y)$ 在 Φ_{ar}^{I} 中由 $A(a,b)$ 公式表示，并有 $Q=\{x\,|\,$有 y，使 $\mathscr{A}(x,y)\}$ 是递归可枚举而非递归的"。事实上，Gödel 在其定理的

证明中，定义了 N 上的二元关系 $\mathscr{B}(x,y)$：

$\mathscr{B}(x,y)$当且仅当"x 是某个 $A\in L^{S_{ar}}$的编码，y 是该 A 在 Φ^{l}_{ar}中证明的编码"

（这种编码也称作 Gödel 数，其是将 Φ^{l}_{ar}中的形式对象，即公式、规则、证明均给以相应的自然数编码。自然，该编码也是满足在第二章中所讲的编码定义的）。且 Gödel 证明：该 $\mathscr{B}(x,y)$ 可由 Φ^{l}_{ar}的一公式 $A(a,b)$来表示。从 Church 1936年的结果知：$P=\{x|$有 y，使 $\mathscr{B}(x,y)\}$是递归可枚举而非递归的集合。那么，由**定理19**，即有 $n\in N$（事实上，$n\in \overline{P}-\{x|\Phi^{l}_{ar}\vdash\neg\exists yA(\underline{x},y)\}$），使

$$\Phi^{l}_{ar}\nvdash\exists yA(\underline{n},y),\neg\exists yA(\underline{n},y)。$$

而 Gödel 所找的 n 为$\forall y\neg A(a,y)$的编码，他所找的 Φ^{l}_{ar}不可判定公式为$\forall y\neg A(\underline{n},y)$，即：

$$\Phi^{l}_{ar}\nvdash\forall y\neg A(\underline{n},y),\quad\neg\forall y\neg A(\underline{n},y)。$$

易知：

$$\Phi^{l}_{ar}\vdash\exists yA(\underline{n},y)\leftrightarrow\neg\forall y\neg A(\underline{n},y)。$$

不过，**定理 19** 也建立了递归不可判定和不完全（即形式不可判定）的关系。

这里，对"完全"、"不完全"再作些直观的描述。一阶逻辑系统的完全性是说：对任 Φ 和 φ，

$$\Phi\models\varphi\quad\Rightarrow\quad\Phi\vdash\varphi；$$

而形式的自然数的算术系统不完全则是说：有 Φ^{l}_{ar}，A，使

$$\Phi^{l}_{ar}\nvdash A,\neg A\quad 或\quad \Phi^{l}_{ar}\not\models A,\neg A。$$

这两者显然并无任何矛盾。前者说的是形式推理系统是足够强的；后者说的却是 Φ^{l}_{ar}没能把和它有关的公式均刻画住，即满足 Φ^{l}_{ar}的模型中，有的可使 A 为真，而有的又可使 A 为假。当然，很自然的一个想法，是增加 Φ^{l}_{ar}中公理的个数，使减少满足这些公理的模型数，以便将有关的任何公式的真假性均能刻画住（或称抓住），比如说，将 A 增加进来，使得到新的公理集 $\Phi^{l}_{ar}\cup\{A\}$，或者，将$\neg A$ 增加进来，使得到另一个新的公理集 $\Phi^{l}_{ar}\cup\{\neg A\}$。但是，Gödel 的

证明方法同样适用于新得到的两个公理系统，即仍然有它们的不可判定的公式。其根本原因是，无论如何增加 Φ^{L}_{ω} 的公理，均不能将自然数刻划到范畴。这样，就会有不同的模型存在（完全不同构），而有区别这些模型的公式，比如说 A，A 在一个模型中为真，而在另一模型中又为假；并且，这 A 还是和 Φ^{L}_{ω} 相关的，即 $A\in L^{S_{\omega}}$。**定理19**则从递归可枚举的角度说明了这一点。

关于 Gödel 不完全定理，目前仍在进行着研究。首先，Kleene 在其有名的专著［33］中，不仅将递归可枚举谓词作了枚举，且将递归可枚举和可证明性联系起来。其次，Smullyan 在其两本论著中[51,52]，不只总结了 Gödel 不完全定理的研究，且以更抽象的形式表述和证明了 Gödel 不完全定理。第三，Kotlarski 在其文章［54，55］中，通过定义增长非常快的函数，而得到其不可证性的结果。第四，最早（1976年）发现数学中的定理而在 PA 中不可证的，是 Paris，Harrington 的结果，这是有关有穷形式 Ramsey 定理的一个推广[53,57]；在1984年，Putnam 来华时曾介绍过 Kripke 关于无穷博奕的一个不可判定命题[56]。第五，杨瑞光在 Kleene 工作的基础上，证得有可数个不可判定的公式存在。即设 $\mathcal{T}_n(z,x_1,\cdots,x_n,y)$ 是表示 Kleene 枚举谓词 $T_n(z,,x_1,\cdots,x_n,y)$（实际上，枚举谓词为 $(Ey)Tn(z,x_1,\cdots,x_n,y)$）的，且能找到 k_0，使对任自然数序组 $(x_1,x_2,\cdots,x_n)$，公式 $\forall y\neg\mathcal{T}_n(\underline{k}_0,\underline{x}_1,\cdots,\underline{x}_{i-1},\underline{k}_0,\underline{x}_{i+1},\cdots,\underline{x}_n,y)$ 不可判定。

2.8　二阶逻辑和 Q 量词逻辑

（1）二阶逻辑 L^{S}_{II}

符号集合 S：增加 n -元谓词变元 $V^n_0,V^n_1,V^n_2,\cdots,n\geqslant1$，用 X,Y 来表示谓词变元。

二阶语言 L^{S}_{II}：增加 S -公式，即如果 $\varphi(R)$ 是 S -公式，则 $\exists X\varphi(X)$ 也为 S -公式。

二阶语义：S -结构 $\mathfrak{A}$ 中的二阶赋值 γ 为映射，$\gamma:\{V^n_i\cdots\}\to\mathscr{R}$，$\mathscr{R}=\bigcup_{i\in\omega}A^i$，$A$ 为 $\mathfrak{A}$ 的论域。且解释 $\mathfrak{I}=(\mathfrak{A},\gamma)$

使

$$\mathfrak{I}\models\underset{(X\text{为}n\text{元})}{\exists X\varphi(X)}$$ 当且仅当有 $C\subset A^n$ 使

$$\mathfrak{I}\frac{C}{X}\models\varphi(X)。$$

1） 对“$\forall X$” 类似进行定义，也可看成是“$\neg\exists X\neg$”的简写。

2）将$\forall x\forall y(x\equiv y\leftrightarrow\forall X(Xx\leftrightarrow Xy))$作为二阶逻辑公理，若记之为$\forall x\forall yE(x,y)$，则可用来代替等词，比如 $a\equiv b$ 即写成 $E(a,b)$。

3）在**2.2例8**中已看到二阶逻辑的描述能力强于一阶的；在**2.7**还介绍了一阶公式无法刻画有限结构（**定理17**），用二阶公式即可克服这一困难。如使用函数变元 g，公式

$\varphi_{fin}=\forall g(\forall x\forall y(gx\equiv gy\longrightarrow x\equiv y)\longrightarrow\forall x\exists y\ x\equiv gy)$，

即表示了有限结构，即 $\mathfrak{I}\models\varphi_{fin}$当且仅当每个1-1映射均满射，当且仅当论域 A 有限。

二阶推导规则：

$(\exists_{\text{右}}^{2})\varphi(R)\vdash\exists X\varphi(X)$，$X$ 可换 $\varphi(R)$中的某些 R，当然也可全换。

$(\exists_{\text{左}}^{2})$若 $\varphi(R)\vdash\psi$，且 ψ 中无 R 出现，则$\exists X\varphi(X)\vdash\psi$。

类似一阶逻辑，可得二阶逻辑的语法紧性：

(a)$\Phi\vdash\varphi$当且仅当有 $\Gamma\subseteq\Phi,\Gamma\vdash\varphi$;

(b)Φ 协调当且仅当任有穷的 $\Phi_0\subseteq\Phi,\Phi_0$均协调。但是，二阶逻辑不具有语义紧性。

定理20 有 Φ，任有穷的 $\Phi_0\subset\Phi$ 可满足，而 Φ 不可满足。

证明 取 Φ 为$\{\varphi_{fin}\}\cup\{\varphi_{\geq n}|n\geq 2\}$，那么，任有穷的 $\Phi_0\subset\Phi$ 均可满足，但 Φ 却不可满足。证毕。

由该定理，立得：

定理21 有 Φ 和 φ，且 $\Phi\models\varphi$，但无 $\Phi\vdash\varphi$。

证明 类似一阶逻辑，二阶逻辑也具有正确性。如果二阶逻辑也具有完全性，则从它的语法紧性立得它的语义紧性；但由**定理**

20，二阶逻辑不具有语义紧性，故其也就不具有完全性。证毕。

定理22 二阶逻辑不满足 Löwenheim-Skolem 定理。

证明 可如下定义出 $\varphi_{unc} \in L_{II}^{S}$，使

$$\mathfrak{I} \models \varphi_{unc} \quad 当且仅当 \quad A 不可数，$$

这样，该 φ_{unc} 即是有模型但无任何可数模型的二阶公式例子。如下构造 φ_{unc}：

(1) 先造公式 $\varphi_{fin}(R)$：

$\forall g[\forall x \forall y(Rx \wedge Ry \wedge Rgx \wedge Rgy \wedge (gx \equiv gy \to x = y)) \to \forall x \exists y x = gy]$，易知：$(\mathfrak{A}, \gamma) \models \varphi_{fin}(R)$ 当且仅当 $R^{\mathfrak{A}}$ 有穷。

(2) 因为任 A 最多可数当且仅当在 A 上有序关系，且 A 中每个元素在该关系下，仅有有限多个前驱。故定义 $\varphi_{\leq ctbl}$：

$$\left.\begin{array}{l}\exists Y(\forall x \neg Yxx \wedge \forall x \forall y \forall z((Yxy \wedge Yyz) \longrightarrow Yxz) \\ \wedge \forall x \forall y(Yxy \vee x \equiv y \vee Yyx) \\ \wedge \forall x \exists X(\varphi_{fin}(X) \wedge \forall y(Xy \leftrightarrow Yyx))\end{array}\right\}$$

(3) φ_{unc} 定义为 $\neg \varphi_{\leq ctbl}$。

定理证毕。

（2）Q 量词逻辑（简称 Q 逻辑）

符号集合 S：对 L^{S}，再加上量词 Q；

Q 逻辑语言 L_{Q}^{S}：若 $\varphi(a)$ 为 S -公式，到 $Qx\varphi(x)$ 也为 S -公式；

Q 逻辑语义：

$$\mathfrak{I} \models Qx\varphi(x) 当且仅当 \{a \in A \mid \mathfrak{I}\frac{a}{x} \models \varphi(x)\} 不可数。$$

1) $\neg Qx(x \equiv x)$ 即刻画了最多可数；

2) 序的公理再加上公式 $\varphi_{0} = (Qx x \equiv x \wedge \forall x \neg Qy < x)$，则它们刻划了不可数的序关系，且其中任一元素在该序关系之下有最多可数个前驱。

3) $Qx(x \equiv x)$ 的模型一定不可数，故从严格的意义上讲，L_{Q}^{S} 不满足 Löwenheim-Skolen 定理；但是有：每个可满足的 $\varphi \in L_{Q}^{S}$，其都有势 $\leq \aleph_{1}$ 的模型。

L_{Q}^{S} 的推理规则：

$(Qx\Rightarrow Qy)$　$Qx\varphi(x)\vdash Qy\varphi(y)$　　　　（约束变元替换），

(Q_+^1)　$\vdash\neg Qx(x\equiv a\vee x=b)$（单个或两个元素非不可数），

(Q_+^2)　$\forall x(\varphi(x)\rightarrow\psi(x))\vdash Qx\varphi(x)\rightarrow Qx\psi(x)$

（有不可数多子集的集合，也不可数），

(Q_-)　$\neg Qx\exists y\varphi(x,y),Qy\exists x\varphi(x,y)\vdash\exists xQy\varphi(x,y)$

（若最多可数多集合之并不可数，

则这些集合中至少有一个不可数）。

类似一阶逻辑，在“$\Phi\models\varphi$当且仅当$\Phi\vdash\varphi$”意义下，L_Q^S是具有正确性和完全性的，只要Φ最多可数的话。详见：H. J. Keisler: logic with the Quantifier “there exist uncountable many”, Annals of Math. logic, **Vol.** 1, 1－93, 1970。

由此，不难得到：

定理23　$\Phi\in L_Q^S$，且Φ可数，则：Φ可满足当且仅当每个有穷的$\Phi_0\subset\Phi$可满足。

但是，设S有不可数多个个体词，且设

$$\Phi=\{\neg c\equiv d\mid c,d\in S,c\neq d\}\cup\{\neg Qx(x\equiv x)\},$$

则不难知道，Φ的每个有穷子集均可满足，但Φ却不可满足。

从上述3)，可以进一步比较L_Q^S和L_1^S的相似之处。

第三节　有穷性逻辑和有穷性数学

上节，较详细地介绍了一阶逻辑和它的局限性，并在末尾引入了二阶逻辑和Q量词逻辑。对这后二种逻辑，虽然有较强的表达能力（或刻划能力），但从其推导规则看，仍无摆脱有穷性原则。本节，即定义并初步研究一下所谓的有穷性逻辑和有穷性数学，从中可望看到它们最本质的特点。

3.1　有穷性逻辑和有穷性数学

设S是有穷的符号表；S^*表示S上的所有有穷长的符号串之集。设$I\subseteq S^*$，且I是递归集。那么，易知：任$Q\subseteq I$，Q递归当且

仅当 Q 和 $I-Q$ 均递归可枚举。

带否定词“$\neg$”的有穷性逻辑 $L^S_\neg$ 为满足如下三条规定的逻辑：

（1）S 为有穷符号集，且 $\neg\in S$；

（2）公式的形成规则是递归的，即其公式集合 $F\subseteq S^*$，且 F 是递归集；

（3）公理集 $A\subseteq F$ 是递归集，推导规则

$$R\subseteq \underbrace{F\times\cdots\times F}_{n}\quad（n\text{ 为自然数}）$$

也是递归集；且设“$\vdash_{L^S_\neg}$”为 $L^S_\neg$ 的形式推导符号，则对任 $f\in F$，均满足：如果有 $\vdash_{L^S_\neg} f$，则没有 $\vdash_{L^S_\neg}\neg f$；且如果有 $\vdash_{L^S_\neg}\neg f$，则没有 $\vdash_{L^S_\neg} f$。

类似地定义 $L^S_\neg$ 的**形式证明**。这里略去。

现在定义有穷性逻辑 $L^S_\neg$ 的有穷性数学。

称 $I\subseteq F$ **是好的**，当且仅当 I 是递归集，且对于任 $f\in F$，均有 $f\in I$ 当且仅当 $\neg f\in I$. 那么显然有：若 I 是好的，则 $f\in I$ 当且仅当 $\neg\neg f\in I$。

$L^S_\neg$ 的**有穷性数学 M**，是 F 的好的子集 F_M，且 M 的公理集 $A_M\subseteq F_M$，且 A_M 是递归集。

称 $L^S_\neg$ 的**有穷性数学 M 协调**，当且仅当对任 $f\in F_M$，不会同时有 $A_M\vdash_{L^S_\neg} f$，且有 $A_M\vdash_{L^S_\neg}\neg f$。

称 $L^S_\neg$ 的**有穷性数学 M 完全**，当且仅当对任 $f\in F_M$，或者有 $A_M\vdash_{L^S_\neg} f$，或者有 $A_M\vdash_{L^S_\neg}\neg f$。

那么有：

定理24 设有穷性数学 M 协调，且设 $I\subseteq F_M$ 是好的，还有 $Q'=\{f\mid A_M\vdash_{L^S_\neg} f, f\in I\}$，不是递归集，则 M 不完全，即至少有一公式 $f\in I$，使既无 $A_M\vdash_{L^S_\neg} f$，也无 $A_M\vdash_{L^S_\neg}\neg f$。

证明 由 M 协调，所以无任何 $f\in I$，使得 $f,\neg f\in Q'$。那么，若能证得：至少有～$f\in I$，但 $f,\neg f\in I\text{-}Q'$，则定理即证明完毕。用反证法证之。若无任何 $f\in I$，使 $f,\neg f\in I\text{-}Q'$，结合 M 是协调的

(即无 $f\in I$,使 $f,\neg f\in Q'$),故 Q' 和 $I-Q'$ 恰有相应的公式，即 $f\in Q'$ 当且仅当 $\neg f\in I\text{-}Q'$,或者是反过来,$\neg f\in Q'$ 当且仅当 $f\in I\text{-}Q'$。这样，从 Q' 是递归可枚举集（这是由于公理和规则的递归性），可知 $I\text{-}Q'$ 也是递归可枚举集，因而，Q' 又是递归集了。这与假设矛盾。故定理证毕。

定理25 设 M、I 同**定理24**中的，且设

$$Q=\{f|A_M\vdash_{L^S_\neg}f,f\in I,\text{且} f \text{的第一符号非“}\neg\text{”}\}$$

不是递归集，则 M 不完全。

证明 令 $R=\{f|A_M\vdash_{L^S_\neg}f,f\in I,\text{且} f\text{的第一符为“}\neg\text{”}\}$

而令 $Q'=Q\cup R$, 易知,$Q\cap R=\phi$,

故 $I-Q=(I-Q'\cup R$,而由 Q 不是递归集，故 $I-Q$ 是递归可枚举集；因而，$I-Q'$ 也非递归可枚举集（否则，∵递归可枚举集对“$\cup$”封闭，由 R 和 $I-Q'$ 均为递归可枚举集，故 $I-Q$ 也为递归可枚举集)。从而，Q' 不是递归集。使用**定理 24** 立得该定理的结果。

不难看到，一阶逻辑 L^S_1，二阶逻辑 L^S_{I}，以及 Q 量词逻辑 L^S_Q 均是有穷性逻辑，且刻划自然数的算术的一阶公理集 Φ^{Sar}_1，自然也就是有穷性数学。且也不难看到，在**定理 19** 的证明中，所找的好的公式集 $I=\{\underbrace{\neg\cdots\neg}_{i}\exists yA(\underline{n},y)|n,i\in N\}$,而 $Q=\{\exists yA(\underline{n},y)|\Phi^{Sar}_1\vdash\exists yA(\underline{n},y),n\in N\}$。

由上述**定理 24** 和**定理 25** 不难理解，只要所给的形式数学足以刻划（用其形式推理关系、语言符号来刻划）一个非递归的集合，则会有形式不可判定的公式存在。事实上，只要试图刻划可数的论域，而所给的公式又足以刻划非递归的集合时，即会有形式不可判定的公式存在。

另方面，从上述结果也可看到：如果协调的有穷性数学 M 完全，则无任何好的 $I\subseteq F_M$，使 $Q=\{f|A_m\vdash_{L^S_\neg}f,f\in I\}$ 不递归。亦即，Q 均递归，即均可判定。可见，对这种有穷性数学，只有不完全的才具有复杂性。

3.2 注记和评论

（1）可以研究有穷性逻辑的语义，即定义解释：$\mathfrak{I}:F\longrightarrow\{0,1\}$；当然要使 $\mathfrak{I}$ 满足对"¬"的非矛盾性。由此，只要所给的语义定义合理，也可讨论完全性问题。作者觉得，满足完全性的有穷性逻辑，用其所表达的有穷性数学，上述所给的不完全性的充分条件，也是必要的。有兴趣的读者可以去研究。

（2）作者觉得，造成"有穷性"的根本所在，是公式的有穷长。而递归化的推理规则又是和这相适应的。因而，才有逻辑的完全性，且有数学的不完全性。故要想克服这种种缺陷，只能允许有无穷长的公式。

在60年代构造的非有穷性逻辑 $L_{\omega_1\omega}$，就在某种意义上克服了上述缺陷。$L_{\omega_1\omega}$不同于 L_1^S 的最本质处，是允许有公式 $\bigwedge\Phi$ 和 $\bigvee\Phi$，其中，Φ 是任意可数的公式集合，而大写的"$\bigwedge$"和"$\bigvee$"，分别表示将这可数个公式"合取"和"析取"起来。比如，公式

$$\forall x(x\equiv 0\vee x\equiv 1\vee x\equiv 2\vee\cdots)$$

即可用公式　　$\forall x\bigvee\Phi$（而 $\Phi=\{x\equiv n\mid n\in N\}$）

来表示之，其正刻划了自然数的算术的论域。又如，公式　$\forall x\bigvee\Phi$（而 $\Phi=\{\underbrace{x\cdot\cdots\cdot x}_{n}\equiv e\mid n\in N\}$）正刻划了挠群元素的有穷阶，再如，

公式 $\Psi_{fin}=\bigvee\{\neg\varphi_{\geqslant n}\mid n\geqslant 2\}$（为同二阶公式相区别，记此公式为 Ψ_{fin}，而不是 φ_{fin}）正刻划了有穷论域的结构。

当然，$L_{\omega_1\omega}$的推理规则也不再满足递归性了。如有规则

（$\bigvee_-$）如果对任 $\varphi\in\Phi$，均有 $\varphi\vdash_{L_{\omega_1\omega}}\Psi$，则有 $\bigvee\Phi\vdash_{L_{\omega_1\omega}}\Psi$；

该（$\bigvee_-$）显然是非递归的。因为要知道是否有

$$\bigvee\Phi\vdash_{L_{\omega_1\omega}}\Psi,$$

必须要对可数的集合 Φ，判断任意一个 $\varphi\in\Phi$，是否均有 $\varphi\vdash_{L_{\omega_1\omega}}\Psi$。而这显然不可能在有穷步完成。，但是，正由于这种规则的非递归性，才与公式的无穷长相适应。我们前边证 L_1^S的完全性定理所用的方法，正是修改了证 $L_{\omega_1\omega}$的完全性定理的方法。也就是说，对

$L_{\omega_1\omega}$是可建立正确而完全的推理系统的。详见［46］。

话又说回来，有穷性逻辑，推理的递归性正是与其相适应的。就是说，在那里，不能建立非递归的推理系统，否则就不满足正确性了。

另请读者注意，关于$L_{\omega_1\omega}$的一切规定和性质，不再有“形式”两个字，因为没有办法像对L_1^S那样，只从公式的形式上，就可判断推理关系的存在与否了。

最后，对二阶逻辑L_1^S，我们再谈几句。前边将L_1^S也算作有穷性逻辑，这是由于其公式和规则均适合有穷性逻辑的定义。但另方面也请注意：二阶公式是可描述非有穷性的内容的，如公式$\forall XXa$，其表示了非可数多个公式。因而，推理规则的递归性与其不相适应（与其描述性），故L_1^S不可能完全。但是，这并不排除增加非递归的规则，而使其得到正确而又完全的推理系统（此时，同样也不可能是“形式”的推理系统了）。

（3）由上述讨论，能否建立一个既包含$L_{\omega_1\omega}$也包含L_1^S在内的，正确而又完全的逻辑系统？看来，这是一个十分重要而又具有相当难度的问题。

第四节　有穷和无穷命题演算

为了在下一章顺利地介绍一般逻辑和一般数学，使读者容易理解和接受；这里，作为准备，着重介绍无穷命题演算。但为了与无穷命题演算相对照，故先介绍有穷命题演算中的一些最重要的内容。

4.1　有穷命题演算

（1）符号：命题词$p_0, p_1, p_2, \cdots$

逻辑连接词$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

结合符号（,）；

（2）公式：任一命题词是公式；若A，B是公式，则$\neg A$，（A

$\wedge B),(A \vee B),(A \to B),(A \leftrightarrow B)$均是公式；仅上述为公式；

(3) 推理规则：如 (**2.3**) 中的$(\in),(\tau),(\neg),(\wedge_+),(\wedge_-),(\vee_+),(\vee_-),(\to_-),(\to_+),(\leftrightarrow_-),(\leftrightarrow_+)$等。

类似地可定义**形式证明**、**语义**，从而研究命题演算的完全性和正确性。留作练习（叙述有关的定义，陈述有关定理，并证明之）。

(4)对命题词作语义解释，即相当于将它们当成取“真”或“假”（即“1”或“0”）的变元，而$\neg p$，p 则是一个变元的函数，$p_1 \wedge p_2$，$p_1 \vee p_2$，$p_1 \to p_2$，$p_1 \leftrightarrow p_2$则均是二个变元的函数（定义域和值域均为 $\{0, 1\}$）。这些统称作**命题函数**。不难知道：

一个变元的命题函数有$2^{2^1}=4$个，即 $F(p)$如右：

p \ $F(p)$	恒真	恒假	p	$\neg p$
1	1	0	1	0
0	1	0	0	1

二个变元的命题函数有$2^{2^2}=16$个，即 $F(p_1,p_2)$：$0, p_1 \wedge p_2, \neg p_1 \wedge p_2, p_1 \wedge \neg p_2, \neg p_1 \wedge \neg p_2, (p_1 \wedge p_2) \vee (\neg p_1 \wedge p_2), (p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2), (p_1 \wedge p_2) \vee (\neg p_1 \wedge \neg p_2), (\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2), (\neg p_1 \wedge p_2) \vee (\neg p_1 \wedge \neg p_2), (p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge \neg p_2), \cdots, (p_1 \wedge p_2) \vee (\neg p_1 \wedge p_2) \vee (p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge \neg p_2)=1$。

……

n 个变元的命题函数有2^{2^n}个，即 $F(p_1,\cdots,p_n)$：$0, p_1 \wedge p_2 \wedge \cdots \wedge p_n, p_1 \wedge \neg p_2 \wedge p_3 \wedge \cdots \wedge p_n, \cdots, \neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n, \cdots, 1$。

对上述这种表达式，还有所谓的**余式定理**：即任

$$F(p_1,\cdots,p_n)=\bigvee_{i_1,\cdots,i_n \in \{0,1\}} \left(F(i_1,\cdots,i_n) \wedge x_1^{i_1} \wedge x_2^{i_2} \wedge \cdots \wedge x_n^{i_n}\right),$$其中，

$$x_j^{i_j}=\begin{cases} x_j, 当\ i_j=1时 \\ \neg x_j, 当\ i_j=0时, \end{cases} j=1,\cdots,n。$$

有兴趣读者请试证明之。

(5) 命题函数和公式之间的关系：如 $p \wedge p$ 和 p，$p_1 \to p_2$和$\neg p_1 \vee p_2$等等，就命题函数言，是同一个命题函数，但就公式而言，

却是不同的。而数学中两个等价命题正说明是同一个命题函数，但却是两个不同的命题（尽管相等价）。而命题逻辑推理系统（或前边的谓词逻辑推理系统）正是证明这些命题的关系，而不是证明命题函数的关系。事实上，命题函数是等值的命题类的总称。

(6)前述命题函数的表示式是命题类中一个很典型的表示式，称作**标准析取范式**。不难知道，当把标准析取范式中的"$\wedge$"和"$\vee$"位置交换之后，所得到的也是命题类中的一个很典型的表示式，称作**标准合取范式**。余式定理则是对任给的命题表示式写成其标准析取范式的一种方法。类似地，也有将任给的命题表示式写成其标准合取范式的方法。请读者试给出。事实上，标准析取范式和标准合取范式是一种对偶形式。

(7) 特别地，我们将 $p_1\wedge\cdots\wedge p_n$ 记作 $\bigwedge\limits_{i=1}^{n}p_i$，而将 $p_1\vee\cdots\vee p_n$ 记作 $\bigvee\limits_{i=1}^{n}p_i$，并称二者分别为 n-元合取和 n-元析取连接词。而当令 $T=\{1,\cdots,n\}$时，则 $\bigwedge\limits_{i=1}^{n}p_i$ 可写为 $\bigwedge\limits_{i\in T}p_i$，而 $\bigvee\limits_{i=1}^{n}p_i$ 可写为 $\bigvee\limits_{i\in T}p_i$。

(8) P_w：这是王浩在50年代末建立的命题逻辑推理系统，其特点是非常简单，而且能**完全机械地**知道任一公式序列和一公式之间是否满足推理关系。正是这一系统的建立，使王浩获了大奖，并被公认为机器定理证明的创始人。

P_w的符号、公式同（1）中的，只是其推理规则如下述：(l_0) **称作公理，而其他称作规则**。

(l_0) 公式序列 Γ_1和 Γ_2称作满足"$\vdash_w$"（记为 $\Gamma_1\vdash_w\Gamma_2$），如果 Γ_1和 Γ_2均为命题词的序列，且有一个命题词 p 同时在 Γ_1和 Γ_2中出现；

($l1a$) $\Gamma_1, A\vdash_w\Gamma_2$ 当且仅当 $\Gamma_1\vdash_w\Gamma_2, \neg A$，其中 Γ_1,Γ_2为公式序列，而 A 为公式（下同，且 B 也为公式）；

($l1b$) $\Gamma_1\vdash_w\Gamma_2, A$ 当且仅当 $\Gamma_1, \neg A\vdash_w\Gamma_2$；

($l2a$) $\Gamma_1\vdash_w\Gamma_2, A$ 且 $\Gamma_1\vdash_w\Gamma_2, B$ 当且仅当 $\Gamma_1\vdash_w\Gamma_2, A\wedge B$；

($l2b$) $\Gamma_1, A, B\vdash_w\Gamma_2$ 当且仅当 $\Gamma_1, A\wedge B\vdash_w\Gamma_2$；

($l3a$) $\Gamma_1 \vdash_w \Gamma_2$，$A$，$B$　当且仅当 $\Gamma_1 \vdash_w \Gamma_2$，$A \vee B$；

($l3b$) Γ_1，$A \vdash_w \Gamma_2$　且 Γ_1，$B \vdash_w \Gamma_2$　当且仅当 Γ_1，$A \vee B \vdash_w \Gamma_2$；

($l4a$)，($l4b$)，($l5a$)，($l5b$) 是有关"$\rightarrow$"和"$\leftrightarrow$"的规则，请读者给予相适应地补齐。

已如前述，(l_0)为公理，而从($l1a$)—($l5b$)这十条规则，从左(右)往右(左)是引入(消去)连接词，故称作相应连接词引入(消去)律。且规则的标记中带有"a"("b")的是对后件(前件)中引入或消去。

形式证明的定义同前，略去。

利用该系统P_w，对任给的公式序列 Γ 和公式 A，均可机械地推知其有无满足推理关系"$\vdash_w$"。办法如下：

先在 Γ 和 A 间加上推理符号"$\vdash_w$"，而后利用($l1a$)—($l5b$)不断地消去连接词，最后得到有穷个如(l_0)中的 Γ_1，Γ_2的形式(Γ_1，Γ_2中均为命题词)，接着判断所得的有穷条(l_0)的形式，若所有各条均满足(l_0)(即至少有一个命题词 p 同时在 Γ_1，Γ_2中出现)，则最初所给的 Γ 和 A 具有推理关系(即 $\Gamma \vdash_w A$)；否则，即有穷条(l_0)的形式中，若有一条不满足(l_0)(即无同一命题词 p 同时在 Γ_1，Γ_2 中出现)，则最初所给的 Γ 和 A 不具有推理关系(即 $\Gamma \not\vdash_w A$)。当 $\Gamma \vdash_w A$ 满足时，把上述过程逆过去，就是它们满足推理关系的形式证明。

例 35　证明$\neg(A \wedge B) \vdash_w \neg A \vee \neg B$

① $\vdash_w \neg A \vee \neg B, A \wedge B$　　(由($l1b$))

(2.1) $\vdash_w \neg A \vee \neg B$，$A$　　(由①，($l2a$) 得二条)

(2.2) $\vdash_w \neg A \vee \neg B$，$B$

(3.1) $\vdash_w A$，$\neg A$，$\neg B$　　(由(2.1)，($l3a$))

(4.1) A，$B \vdash_w A$　　(由(3.1)，($l1a$))

(3.2) $\vdash_w B$，$\neg A$，$\neg B$　　(由(2.2)，($l3a$))

(4.2) A，$B \vdash_w B$　　(由(3.2)，($l1a$))

⑤ 由 (4.1) 和 (4.2) 均满足(l_0)，故证明，且这过程逆过去，就是

所需的形式证明。

例36 证明$\vdash_{\varpi}((p\vee\neg q)\wedge q)\leftrightarrow(p\wedge q)$。留作练习。

4.2 无穷命题演算*

这小节实际是在介绍无穷命题函数，而在附注中则补充介绍了无穷命题演算。请读者注意。

(1)无穷命题函数、全称和存在连接词

---

*对无穷命题演算（这里不称作命题函数，请注意），也可建立其正确和完全的推理系统，如下。

(1) 符号：命题词集合 P；

$\neg$，$\wedge$，$\vee$，逻辑连接词（注意不同于有穷命题演算中的$\wedge$，$\vee$）；

（ ， ），结合符号；

(2) 公式：任一命题词 p（$\in P$）是公式；

若 X 为公式，则$\neg X$ 为公式；

若 X 为公式集合，则$\wedge X$，$\vee X$ 均为公式；公式仅由上述得到。

(3) 推理规则：有类似于有穷命题演算的（$\in$）（τ）和（$\neg$）的，记为（$\in'$），（τ'）和（$\neg'$），只是要将那里出现的"公式序列"均改为"公式集合"；此外，再加上$(\vee_+)$，$(\vee_-)$，$(\wedge_+)$，$(\wedge_-)$，如下

$(\wedge_+)\Phi\vdash\wedge\Phi$，$\Phi$ 为公式集；

$(\wedge_-)\wedge\Phi\vdash\varphi$，$\Phi$ 为公式集，$\varphi\in\Phi$；

$(\vee_+)\varphi\vdash\vee\Phi$，$\Phi$ 为公式集，有 $\varphi\in\Phi$；

$(\vee_-)$若 $\Phi,\varphi\vdash B$，对任 $\varphi\in\Phi'$，

则 $\Phi,\vee\Phi'\vdash B$，Φ,Φ' 均为公式集，而 B 为一公式；

(4) 类似地定义证明（注意，这里也无"形式"二字作定语）、语义。

(5) 仅以证完全性为例，即：

对任公式集 Φ，公式 A，若有 $\Phi\models A$，则有 $\Phi\vdash A$。(证明方法仿照有穷命题演算)

引理 设 A 中出现的所有命题词 $P\in T$，且对任给的语义解释 I，对任 $p\in T$，

令 $$p'=\begin{cases}p,\text{当 }I(p)=1\text{时，}\\ \neg p,\text{当 }I(p)=0\text{时；}\end{cases}$$

那么，作$A=\{p'\mid p\in T\}$，

则有：(1)若 $I(A)=1$，则$\wedge A\vdash A$，

(2)若 $I(A)=0$，则$\wedge A\vdash\neg A$。

证明 施归纳于 A 的构造：

1) A 为命题词：显然成立；

2) 归纳：

①若 $A=\neg B$，且已证明(1)$I(B)=1\Rightarrow\wedge B\vdash B$

(2)$I(B)=0\Rightarrow\wedge B\vdash\neg B$；

设 T 是命题的指标集，且 T 的势为 α，那么，α-元命题函数 $F(p_t:t\in T)$ 有 2^{2^α} 种（注意：这里的记号 2^α 仅是一种记法，不像有穷命题演算那里的 2^n）。特别地，α-元**全称连接词** $\bigwedge_{t\in T}$ 定义为

$\bigwedge_{t\in T} p_t=1$　当且仅当对任何 $t\in T$，$p_t=1$；而 α-元**存在连接**

那么，若 $I(A)=1$，则 $I(B)=0$，故 $\wedge B\vdash \neg B=A$；

而 $A=B$，故 $\wedge A\vdash A$；

若 $I(A)=0$，则 $I(B)=1$，故 $\wedge B\vdash B\vdash\neg\neg B=\neg A$；

而 $A=B$，故 $\wedge A\vdash\neg A$；

②若 $A=\wedge B$，且对任 $b\in B$，已证明

$$(1)\ I(b)=1\Rightarrow\wedge b\vdash b;$$

$$(2)\ I(b)=0\Rightarrow\wedge b\vdash\neg b;$$

那么，由于 A 中命题词和 B 中的命题词相同，故　$A=\bigcup_{b\in B} b$，分下述二种情况来证结果：

若 $I(A)=1$，则对任 $b\in B$，$I(b)=1$，故 $\wedge b\vdash b$（对任 $b\in B$）；

$$\text{而}\wedge A=\wedge(\bigcup_{b\in B} b)\overset{(\wedge_-)}{\vdash} c\begin{pmatrix}c\in b,\\ \text{任 } b\in B\end{pmatrix}\overset{(\wedge_+)}{\vdash}\wedge b\overset{(1)}{\vdash} b(b\in B)\overset{(\wedge_+)}{\vdash}\wedge B=A;$$

若 $I(A)=0$，则有 $b\in B$，$I(b)=0$，故 $\wedge b\vdash\neg b$；

$$\text{而}\wedge A=\wedge(\bigcup_{b\in B} b)\overset{(\wedge_-)}{\vdash} c\begin{pmatrix}c\in b\\ \text{任 } b\in B\end{pmatrix}\overset{(\wedge_+)}{\vdash}\wedge b\overset{(2)}{\vdash}\neg b(b\in B)\overset{(\wedge_+)}{\vdash}\vee\{\neg b|b\in B\}\underset{\text{德·摩根}}{\overset{\text{广义}}{\vdash}}\neg(\wedge B)=\neg A;$$

③若 $A=\vee B$，类似可证，有兴趣者请练习。至此，引理证毕。

定理　对任公式 A，若 $\vDash A$，则 $\vdash A$。

证明　设 A 中的命题词 $p\in T$，

从 T 中任选一 p，设 $A_{-p}=A-\{p'\}$；　　　　（∗）

（1）作一语义解释 I，使 $I(p)=1$，那么，相应于该解释，由引理，可得

$$(\wedge A_{-p})\wedge p\vdash A\text{，故 }\wedge A_{-p},p\vdash A;$$

（2）作另一语义解释 I'，使 $I'(p)=0$，那么，相应于该解释，由引理，可得

$$(\wedge A_{-p})\wedge(\neg p)\vdash A\text{，故 }\wedge A_{-p},\neg p\vdash A;$$

由（1）和（2），使用（$\vee_-$），得 $\wedge A_{-p}$，$\vee\{p,\ \neg p\}\vdash A$；但显然可证：$\vdash\vee\{p,\ \neg p\}$，故得 $\wedge A_{-p}\vdash A$。使用同样的方法，可消去 A 中的各 p'，因而得到　$\vdash A$，证毕。（注意：在（∗）那里及后来逐次消去 A 中的各 p'，使用了选择公理。详见第五章的集合论部分）。

由该定理，立刻可得无穷命题演算的完全性定理。

词$\vee_{t\in T}$定义为

$$\vee_{t\in T}\, p_t = 1 \quad 当且仅当有\ t\in T,使\ p_t = 1。$$

（2）无穷命题函数的标准析取范式。

定理26 设 T 是命题的指标集，则任何命题函数 $F(p_t:t\in T)$均可表示为

$$F(p_t:t\in T) = \vee_{s\in C}[(\wedge_{t\in S}p_t)\wedge(\wedge_{t\in(T\text{-}S)}\neg\, p_t)],$$

其中,$C\subseteq P(T)$,$P(T)$表示 T 的幂集。且特别规定,恒取0为值的命题函数表示为 $C=\phi$。

在未证明本定理前,先将有穷命题函数的标准析取范式的结果和该定理相比较。事实上,这里的 C 相当于那里的对2^n个合取式选取时,所选到的部分;那里共有2^{2^n}种不同的选择,而这里 $P(T)$ 中不同的 C 又有2^{2^n}个。另外,这里的 S 相当于那里的每个合取式中那些不加"$\neg$"的命题变元的下标集,而 $T\text{-}S$ 恰好相当于那里的每个合取式中,那些加"$\neg$"的命题变元的下标集。由此可见对该定理的称呼是相宜的。

现在,严格证明该定理。

由 $F(p_t:t\in T)$是$\{0,1\}$上的命题函数，故对自变量 $p_t(t\in T)$ 的任何一组取值,因变量 $F(p_t:t\in T)$均有一个相应的取值。那么,可能有下述三种情况：

1）对 $p_t(t\in T)$的任何一组取值，$F(p_t:t\in T)$均取0为值：此时，即相当于定理中的 $C=\phi$;

2）对 $p_t(t\in T)$的任何一组取值，$F(p_t:t\in T)$均取1为值：此时，即相当于定理中的 $C=P(T)$;

3）对 $p_t(t\in T)$的某些组取值，$F(p_t:t\in T)$取1为值，而对 $p_t(t\in T)$的另外一些组的取值，$F(p_t:t\in T)$取0为值：此时，不妨设对 $p_t(t\in T)$的一组取值，使 $F(p_t:t\in T)$为1，那么，对该组 p_t 的取值，令

$$g(p_t) = \begin{cases} p_t,若\ p_t = 1\ 时 \\ \neg\, p_t,若\ p_t = 0\ 时, \end{cases} t\in T,$$

则显然有 $\bigwedge_{t\in T} g(p_t)=1$；

对该组 p_t 的取值，自然也可分成两部分，即使得 $g(p_t)=p_t$ 的（此时 $p_t=1$），和使得 $g(p_t)=\neg p_t$ 的（此时 $p_t=0$），且设 $S=\{t\mid g(p_t)\equiv p_t, t\in T\}$，则自然有 $T-S=\{t\mid g(p_t)=\neg p_t, t\in T\}$，因而，有

$$\begin{aligned}\wedge_{t\in T} g(p_t) &= (\wedge_{t\in S} g(p_t)) \wedge (\wedge_{t\in T-S} g(p_t)) \\ &= (\wedge_{t\in S} p_t) \wedge (\wedge_{t\in T-S} \neg p_t),\end{aligned}$$

这样，对使 $F(p_t:t\in T)$ 取1的每组 $p_t(t\in T)$ 的取值，即得到一相应的 S；令 C 是所有这些 S 构成之集，则显然有 $C\subseteq P(T)$，且

$$F(p_t:t\in T)=\vee_{S\in C}[(\wedge_{t\in S} p_t)\wedge(\wedge_{t\in T-S}\neg p_t)]。$$

故**定理 26** 证毕。

（3）与**定理 26** 相关的结果：

由**定理 26**，显然，任一 $F(p_t:t\in T)$ 均相对应着唯一的 $C\subseteq P(T)$。

例如，$\wedge_{t\in T} p_t$ 所相应的 $C=\{T\}$，而

$\vee_{t\in T} p_t$ 所相应的 $C=P(T)-\{\phi\}$；

恒真命题函数 $F(p_t:t\in T)=1$ 所相应的 $C=P(T)$；

恒假命题函数 $F(p_t:t\in T)=0$ 所相应的 $C=\phi$。

这样，设 $F(p_t:t\in T)$ 相对应的唯一的 $C\subseteq P(T)$，则将 $F(p_t:t\in T)$ 记为 $F_C(p_t:t\in T)$。那么，易知：

$F_{C_1}(p_t:t\in T)\vee F_{C_2}(p_t:t\in T)$ 即 $F_{C_1\cup C_2}(p_t:t\in T)$，

$\neg F_C(p_t:t\in T)$ 即 $F_{\overline{C}}(p_t:t\in T)$，而 $\overline{C}=p(T)-C$，

$F_{C_1}(p_t:t\in T)\wedge F_{C_2}(p_t:t\in T)$ 即 $F_{C_1\cap C_2}(p_t:t\in T)$，

$F_{C_1}(p_t:t\in T)\rightarrow F_{C_2}(p_t:t\in T)$ 即 $F_{\overline{C}_1\cup C_2}(p_t:t\in T)$，

$F_{C_1}(p_t:t\in T)\leftrightarrow F_{C_2}(p_t:t\in T)$ 即 $F_{(\overline{C}_1\cup C_2)\cap(\overline{C}_2\cup C_1)}(p_t:t\in T)$

亦即 $F_{(\overline{C}_1\cap\overline{C}_2)\cup(C_1\cap C_2)}(p_t:t\in T)$；

故 1）$F_{C_1}(p_t:t\in T)\rightarrow F_{C_2}(p_t:t\in T)$ 恒真，当且仅当

$F_{\overline{C}_1 \cup C_2}(p_t : t \in T)$ 恒真，当且仅当

$\overline{C}_1 \cup C_2 = P(T)$ 当且仅当 $C_1 \subseteq C_2$；

2) $F_{C_1}(p_t : t \in T) \leftrightarrow F_{C_2}(p_t : t \in T)$ 恒真，当且仅当

$F_{(\overline{C}_1 \cup \overline{C}_2) \cup (C_1 \cap C_2)}(p_t : t \in T)$ 恒真，当且仅当

$(\overline{C}_1 \cap \overline{C}_2) \cup (C_1 \cap C_2) = P(T)$，当且仅当 $C_1 = C_2$；

（也可从(1)，得 $C_1 \subseteq C_2$ 且 $C_2 \subseteq C_1$，故 $C_1 = C_2$）；

(4) 设 R 为 T 上的关系（也称 T 上的谓词），那么，对任给 $t \in T$，$R(t)$ 或为1，或为0，所以把 $R(t)$ 可看成命题词 Pt，而 T 为命题 p_t 的指标集。这样，由 R 产生的复合关系 $F(R)$ 定义为 $F(R(t) : t \in T)$，有时，也记作 $F(t;R)(t \in T)$；

而当 $T = T_1 \times T_2 \times \cdots \times T_n$ 时，$F(R)$ 也记作 $F(t_1, \cdots, t_n; R)$ $(t = (t_1, \cdots, t_n) \in T)$。

第四章　一般逻辑和一般数学

上章，研究了有穷性逻辑和有穷性数学，并指出“有穷性”所带来的不足。作为对比，在 **3.2** 节还简单介绍了无穷逻辑 $L_{\omega_1\omega}$。上章末尾，还介绍了有穷命题演算和无穷命题演算，这就为本章的讨论打下了基础。一般逻辑（记作 $\widetilde{\mathscr{L}}$）和一般数学（记作 $\widetilde{M}\mathscr{L}$）是胡国定等近几年的研究结果。它们与第三章所讨论的种种逻辑，如 L_{I}^S，L_{II}^S，L_Q和 $L_{\omega_1\omega}$的最大不同是：不再将语法和语义分开，并且可随意应用 Cantor 集合论。而这些特点，正是除数理逻辑之外的各门数学所遵守的。本章还将进一步比较和分析它们的同异。

第一节　一般逻辑和一般数学的定义

上一章在介绍无穷命题演算后，引入所谓的复合关系。现在简单重述一下。

设 R 为 T 上的关系（也称作 T 上的谓词），那么，由 R 产生的**复合关系 $F(R)$定义为**

$$F_C(R(t):t\in T)\text{（也有时记为 }F_C(t;R),t\in T),C\subseteq P(T)；$$

而当 $T=T_1\times\cdots\times T_n$时，$F(R)$也记作

$$F_C(t_1,\cdots,t_n;R)(t=(t_1,\cdots,t_n)\in T),C\subseteq P(T)。$$

当然，还可设 R 为 $T\times P(T)$上的关系。那么，由 R 产生的**复合关系 $F(R)$定义为**

$$F_C(R(t,\mathscr{S}):(t,\mathscr{S})\in T\times P(T)),C\subseteq P(T\times P(T)),$$

有时也记作 $F_C(t,\mathscr{S};R),((t,\mathscr{S})\in T\times P(T))C\subseteq P(T\times P(T))$；但是，当从另一角度看时，即当认为 R 是

$$T'=T\times P(T)$$

上的关系时，那么，由 R 产生的复合关系 $F(R)$定义为 $F_C(R(t')$：

$t' \in T'$), $C \subseteq T'$。这又归到前述情况了。

1.1 一般逻辑 $\tilde{\mathscr{L}}$ 的定义

设 X，T 为任给的非空集。$\tilde{\mathscr{L}}$ 由下述三部分构成：

(1) 对任意的 X，有连接词的集合

$\tilde{\mathscr{L}}_X = \{F_C \mid C \subseteq P(X)\}$；

(2) 对任意的 X，任意的 T，公式的集合

$\tilde{\mathscr{L}}_F = \tilde{\mathscr{L}}_M \cup \tilde{\mathscr{L}}_L$，

而 $\quad \tilde{\mathscr{L}}_M = \{F_C(R) \mid F_C \in \tilde{\mathscr{L}}_X, R$ 为 X 上的关系$\}$

称作数学公式集，

$$\tilde{\mathscr{L}}_L = \{F_D(p_t; t \in T) \mid D \subseteq P(T)\}$$

称作逻辑公式集。

(3) 对任意的 T，一般推理规则集

$$\tilde{\mathscr{L}}_G = \{F_{D_1}(p_t; t \in T) \Vdash F_{D_2}(p_t; t \in T) \mid D_1 \subseteq D_2 \subseteq P(T)\},$$

这里的“任意”，是为包括各种公式（即不单单是无穷命题函数的标准析取范式），及包括所有可能的推理。

1.2 一般数学 $\tilde{M}$ ($\tilde{\mathscr{L}}$) 的定义

设 X，T 为任给的非空集。一般逻辑 $\tilde{\mathscr{L}}$ 中的一般数学 $\tilde{M}(\tilde{\mathscr{L}})$ 包括下述五部分：

(1) 对给定的 X，X 上的关系 R，由 X 所决定的连接词集合 $\tilde{\mathscr{L}}_X = \{F_C \mid C \subseteq P(X)\}$，且称 $R(x)(x \in X)$ 为原子公式；

(2) 对给定的 X，数学公式集合 $\tilde{\mathscr{L}}_M = \{F_C(R) \mid F_C \in \tilde{\mathscr{L}}_X\}$；

(3) 对任意的 T，逻辑公式集合 $\tilde{\mathscr{L}}_L = \{F_D(p_t; t \in T) \mid D \subseteq P(T)\}$；

(4) 数学公理集 $A_M \subseteq \tilde{\mathscr{L}}_M$；

(5) 对任意的 T，一般推理规则集

$\widetilde{\mathscr{L}}_G=\{F_{D_1}(p_t:t\in T)\Vdash F_{D_2}(p_t:t\in T)\mid D_1\subseteq D_2\subseteq P(T)\}$。

现在举例说明前述定义。

例1 证明 $p_1\wedge p_2\Vdash p_1\vee p_2$

此时，取 $T=\{1,2\}$。

按定理 26 的证明，必有 $D_1,D_2\subseteq P(T)$，使

$$p_1\wedge p_2=F_{D_1}(p_t:t\in T),p_1\vee p_2=F_{D_2}(p_t:t\in T);$$

且∵$p_1\wedge p_2=1$ 当且仅当 $p_1=1,p_2=1$，∴$D_1=\{\{1,2\}\}$；

又∵$p_1\vee p_2=1$ 当且仅当 $p_1=1$，或$p_2=1$，或 $p_1=p_2=1$，∴$D_2=\{\{1\},\{2\},\{1,2\}\}$；

显然 $D_1\subseteq D_2$，故得证。

例 2 证明 $p_1\wedge p_2\Vdash p_2$。

此时，取 $T=\{1,2\}$，由上题，$p_1\wedge p_2$相应的 $D_1=\{\{1,2\}\}$，而p_2相应的 $D_2=\{\{2\},\{1,2\}\}$，显然，$D_1\subseteq D_2$，故证毕。

例 3 证明 $p_{t_0}\Vdash \bigvee\limits_{t\in T}p_t,t_0\in T$。

T 已选定，且 $t_0\in T$。

∵$p_{t_0}=1$ 当且仅当 $p_{t_0}=1$，或 $p_{t_0}=1$且任 $p_t=1,(t\in T)$，故

p_{t_0}相应的 $D_1=\{\{t_0\},\{t_0\}\cup A\mid A\subseteq T\}=\{\{t_0\}\cup A\mid A\subseteq T\}$，

而我们已知，$\bigvee\limits_{t\in T}p_t$ 相应的 $D_2=P(T)-\{\phi\}$，显然，$D_1\subseteq D_2$，故证毕。

例 4 证明 $p_1\wedge(p_1\to p_2)\Vdash p_2$。

此时，取 $T=\{1,2\}$。

∵$p_1\wedge(p_1\to p_2)=1$当且仅当$p_1=1$且 $p_2=1$，故它相应的 $D_1=\{\{1,2\}\}$；

而 $p_2=1$ 当且仅当 $p_2=1$，或$p_1=1$且 $p_2=1$，故 p_2相应的 $D_2=\{\{2\},\{1,2\}\}$；

显然，$D_1\subseteq D_2$，故证毕。

例 5 证明 $p_1\wedge\neg p_1\Vdash p_2$。

此时，选 $T=\{1,2\}$。

∵$p_1\wedge\neg p_1$不可能取1，故它相应的 $D_1=\phi$；

而 p_2 相应的 $D_2=\{\{2\},\{1,2\}\}$；

显然，$D_1\subseteq D_2$，故证毕。

例 6 证明 $p_1 \Vdash p_2 \vee \neg p_2$。

此时，取 $T=\{1,2\}$。

$\because$ p_1相应的 $D_1=\{\{1\},\{1,2\}\}$；而 $p_2\vee\neg p_2$相应的 $D_2=P(T)$；故显然有 $D_1\subseteq D_2$，证毕。

例 7 证明$\neg(\neg(p_0\rightarrow\neg p_1)\rightarrow\neg p_2)\Vdash p_0\rightarrow\neg(p_1\rightarrow\neg p_2)$。

此时，选 $T=\{0,1,2\}$。

由第三章**定理 26** 证明后的讨论知：

$\neg(\neg(p_0\rightarrow\neg p_1)\rightarrow\neg p_2)$相应的 $D_1=\overline{(\overline{\overline{(\overline{D'_0}\cup\overline{D'_1})}}\cup\overline{D'_2})}=\overline{(\overline{D'_0}\cup\overline{D'_1}\cup\overline{D'_2})}=D'_0\cap D'_1\cap D'_2=\{\{0,1,2\}\}$，其中，$D'_i$是 p_i相应的集合；下面同；而 $p_0\rightarrow\neg(p_1\rightarrow\neg p_2)$相应的 $D_2=\overline{D'_0}\cup\overline{(\overline{D'_1}\cup\overline{D'_2})}=\overline{D'_0}\cup(D'_1\cap D'_2)=\{\phi,\{1\},\{2\},\{1,2\}\}\cup\{\{1,2\},\{0,1,2\}\}=\{\phi,\{1\},\{2\},\{1,2\},\{0,1,2\}\}$；

故显然有，$D_1\subseteq D_2$，证毕。

例 8 群就是一个一般数学 $\tilde{M}(\tilde{\mathscr{L}})$(即一般群)。

因为，按定义，如下可见：

(1) 给定的 X，即为群的元素的任意 n 序组($n=1,2,3,\cdots$)，亦即

$X=G\cup G^2\cup G^3\cup\cdots$，其中，$G$ 为群的元素集；X 上的关系 R，即对任 $x\in X$，$R(x)$当且仅当

$x\in G^3\subseteq X$，且 $x=(x_1,x_2,x_3)$，$x_i\in G(i=1,2,3)$，

且 $x_1\cdot x_2=x_3$，而这里的“$\cdot$”为 G 的代数运算；由 X 决定的连接词的集合 $\tilde{\mathscr{L}}_X=\{F_C|C\subseteq P(X)\}$；且对任意的 x_1^0，x_2^0，$x_3^0\in G\subseteq X$，$R(x_1^0, x_2^0, x_3^0)$ 称为 $\tilde{M}(\tilde{\mathscr{L}})$ 的原子公式（或说 $R(x)$为原子公式，而 $x\in G^3\subseteq X$）。

(2) 该 $\tilde{M}(\tilde{\mathscr{L}})$的数学公式集 $\tilde{\mathscr{L}}_M=\{F_C(R)|F_C\in\tilde{\mathscr{L}}_X\}$，其即为群 G 所讨论的全部命题，比如，群的公理（有关封闭性）

$\bigwedge_{x_1\in G}\bigwedge_{x_2\in G}\bigvee_{x_3\in G}R(x_1,x_2,x_3)$（或 $\bigwedge_{(x_1,x_2)\in G^2}\bigvee_{x_3\in G}R(x_1,x_2,x_3)$）它相应的 $C_1=\{G\}\times\{G\}\times(P(G)-\{\phi\})\subseteq P(X)$；又比如，群的公理（有关结合律）

$$\bigwedge_{x_1\in G}\bigwedge_{x_2\in G}\bigwedge_{x_3\in G}\bigvee_{x_4\in G}\bigvee_{x_5\in G}\bigvee_{x_6\in G}(R(x_1,x_2,x_4)\wedge R(x_4,x_3,x_6)\wedge$$

$R(x_2,x_3,x_5)\wedge R(x_1,x_5,x_6))$（或 $\bigwedge_{(x_1,x_2,x_3)\in G^3}\bigvee_{(x_4,x_5,x_6)\in G^3}(R(x_1,x_2,x_4)\wedge R(x_4,x_3,x_6)\wedge R(x_2,x_3,x_5)\wedge R(x_1,x_5,x_6)))$

它相应的 $C_2=\{G\}^3\times(P(G)-\{\phi\})^3$。

(3)，(5)不用另述。

(4) 该 $\widetilde{M}(\widetilde{\mathscr{L}})$ 的公理集 A_M 即群的任一组公理，如第三章例 3 中所选的群的公理，即除上述二条外，尚有另外二条公理，这里略去。

第二节 一般逻辑 $\widetilde{\mathscr{L}}$ 和一般数学 $\widetilde{M}(\widetilde{\mathscr{L}})$ 的解释

$\widetilde{\mathscr{L}}$ 和 $\widetilde{M}(\widetilde{\mathscr{L}})$ 的解释，是随所给的 X 的不同而不同，且也随 X 上的关系 R 的不同而不同。但只要这二者给定了，其他也就相应地定了。详细情况如下。

2.1 公式和推理规则的解释

X 解释为不同数学的对象域，或称论域。

关系 R 解释为 X 的子集 $\hat{R}$，原子公式 $R(x)$ 解释为 X 上的命题，即 $R(x)=\begin{cases}1, & x\in\hat{R};\\0, & x\in X-\hat{R}.\end{cases}$

连接词 $F_C\in\widetilde{\mathscr{L}}_X(C\subseteq P(X))$ 解释为 X-元（或 $|X|$-元）的命题连接词，公式 $F_C(R)$ 解释为 $|X|$-元复合关系，这里，$|X|$ 表示 X 的势，下同。

逻辑公式 $F_D(p_t:t\in T)(D\subseteq P(T))$ 解释为 $|T|$-元复合命题。

一般推理规则“$F_{D_1}(p_t:t\in T)\Vdash F_{D_2}(p_t:t\in T)(D_1\subseteq D_2\subseteq$

$P(T))$”解释为恒真公式

$$“F_{D_1}(p_t:t\in T)\rightarrow F_{D_2}(p_t:t\in T)(D_1\subseteq D_2\subseteq P(T))”。$$

2.2 $\tilde{M}$ $(\mathscr{L})$ 的结构

称 X 的解释和关系 R 的解释为 $\tilde{M}(\tilde{\mathscr{L}})$的结构。常记为 $\mathscr{A}$。在 $\mathscr{A}$ 中定义语义推演关系：

对任何原子公式 $R(x)$；$\mathscr{A}\models R(x)$　当且仅当 $x\in\hat{R}$(或记作 $\hat{R}(x)=1$)；

而对非原子公式 $F_C(R)$，$\mathscr{A}\models F_C(R)$当且仅当 $F_C(R)=1$。

由此，使用命题函数的定义，显然有：对任给公式 $\varphi(\in\tilde{\mathscr{L}}_M)$，当将它解释为命题时，其或者为0，或者为1，亦即 $\mathscr{A}\models\varphi$　或者 $\mathscr{A}\models\neg\varphi$。

2.3 $\tilde{M}$ $(\tilde{\mathscr{L}})$ 的模型

$\tilde{M}(\tilde{\mathscr{L}})$的结构 $\mathscr{A}$ 称作其模型，如果对 $\tilde{M}(\mathscr{L})$的任何公理 $\varphi\in A_M$，均有

$$\mathscr{A}\models\varphi;此时，也记作\ \mathscr{A}\models A_M。$$

称 $\tilde{M}(\tilde{\mathscr{L}})$的公式集合 $\Psi\subseteq\tilde{\mathscr{L}}_M$，和公式 $\varphi\in\tilde{\mathscr{L}}_M$满足语义推演关系，或称 Ψ 能语义地推演出 φ，记作 $\Psi\models\varphi$，如果对 Ψ 的任何模型 $\mathscr{A}$（即有 $\mathscr{A}\models\Psi$)，也是 φ 的模型（亦即 $\mathscr{A}\models\varphi$)。

在不致引起混淆时，常将 $A_M\models\varphi$，简写作$\models\varphi$；φ 也称作 $\tilde{M}(\tilde{\mathscr{L}})$ 的语义定理。

例 9（续例 8）　任给一个具体的群，都是例 8的一般群的一个模型。显然，也可给出上述群的一个结构，但它却不为上述群的模型（留作练习)。在群的任一结构中（包括群的模型在内），群的任一公式，自然要么为真，要么为假；但对一般群而言，可以有群的公式 φ，φ 的真、假性未定。然而，对一般群的任何定理，在群的任一模型中均是成立的。

2.4 一般数学 $\tilde{M}(\mathscr{L})$ 的语义完全性

称一般数学 $\tilde{M}(\mathscr{L})$ 是具体的，如果 $\tilde{M}(\mathscr{L})$ 的模型，从同构意义上讲，是唯一的话。

那么，显然有：

定理 任何具体的一般数学 $\tilde{M}(\mathscr{L})$ 均是语义完全的，即对任 $\varphi\in\tilde{\mathscr{L}}_M$，或者有 $\models\varphi$，者有 $\models\neg\varphi$。

证明 因为从2.2 的讨论知，在 $\tilde{M}(\mathscr{L})$ 的结构中，总会有 $\models\varphi$，或者有 $\models\neg\varphi$；而 $\tilde{M}(\mathscr{L})$ 又只有一个模型，故定理的结论总成立。

例10 自然数的算术是一般数学 $\tilde{M}(\mathscr{L})$。

因为，给定的 X，可取为四个运算和"自然数"的任意 n-序组（$n=1,2,3,\cdots$），即

$$X=\{a,b,c,e\}\times\bigcup_{i=1}^{\infty}N^i,$$

其中，$\{a,b,c,e\}$ 表示有四个关系，而 N 为自然数集合。而 X 上的 R，即为：对任 $(i,t)\in X$，$R(i,t)$ 当且仅当

1）或者 $i=a,t=(x_1,x_2)\in X$，且 $x_1+1=x_2$；

2）或者 $i=b,t=(x_1,x_2,x_3)\in X$，且 $x_1+x_2=x_3$；

3）或者 $i=c,t=(x_1,x_2,x_3)\in X$，且 $x_1\cdot x_2=x_3$；

4）或者 $i=e,t=(x_1,x_2)\in X$，且 $x_1=x_2$。

该 $\tilde{M}(\mathscr{L})$ 的其他各项，略去。

因为该一般数学 $\tilde{M}(\mathscr{L})$ 是具体的，故其也是语义完全的。

第三节 总结和讨论

在第三章的 3.2 小节，我们曾对有穷性逻辑和有穷性数学作了注记和评论，且引出无穷逻辑 $L_{\omega_1\omega}$。现在，我们对第三章和第四章的内容（这全是有关数学逻辑的）作一总结，并作一简单讨论。

第三、四两章中讨论过的数学逻辑计有

$L_1^S, L_1^S, L_Q^S, L_{\neg}^S$ $, L_{\omega_1\omega}, P_W, \widetilde{\mathscr{L}}, \widetilde{M}(\widetilde{\mathscr{L}})$等。其中，$P_W$是最简单的命题逻辑（或更确切地说，是一种有穷命题演算）。我们就先从P_W讲起。可以说，上述任何一种逻辑，均包含P_W在内。和P_W完全相当的是有穷命题演算。第三、四两章中，均讨论或谈起了命题函数，其与命题演算的关系是：命题函数将相应的命题演算的公式，按语义等价与否分成了等价类，而命题函数正可以看作是这等价类的代表（请注意，这里的命题函数、命题演算既包括有穷的，也包括无穷的在内）。在第三章第四节中还讨论了无穷命题函数，并作为附注，证明了无穷命题演算的完全性。按说，无穷命题演算是最一般的逻辑了，即其抽象性和概括性是最强的。但正是由于这种抽象性，使其失去了研究命题本身的结构之可能。在P_W中，引入形式推理符号“$\vdash_W$”，其他命题演算中的推理符号为“$\vdash$”。请注意，在无穷命题演算中，是不能将“$\vdash$”作为“形式”的推理符号看待的。在各逻辑演算中，均以“$\models$”作为其语义推演符号。

现在总结一下介绍得最多的逻辑演算 L_1^S。L_1^S 称作谓词演算、狭谓词演算，或一阶谓词演算等。一般的逻辑书中，大多是只介绍L_1^S。我们在第三章第一节、第二节重点地介绍了L_1^S。具体说来，分成六部分作了研究。(1)符号集$A_S=A\cup S$。其中A称逻辑符号集，而S称数学符号集。(2) 公式集合 $L^S\subseteq A_S^*$。其是由A_S中的符号，按形成规则组合成的任意有穷长的字的集合。(3)语义。即按一种统一的方法解释 L^S中的元素，使与数学中的命题相对应。如用数学符号写，即给出一语义映射 $I:L^S\to\{0,1\}$。当然，该I有一些统一的很自然为人们接受的性质。(4)推理语法。包括推理规则和形式证明。前者即是按某种“组合”规则确定，如何从已知的公式得到它们的推论，或者是结论；且这种“组合”规则完全是“形式”的，即完全从公式的“样子”就可得到它们的推论。而后者则是指如何有限次使用这种推理规则，得到所谓的形式证明，或者说，是其最后一个结论的形式证明过程。(5) 正确性和完全性。即先给出自然的标准，而后证明上述 (3) 和 (4) 二者是相“等价”的。亦就是说，按上述形式证明的过程，证得数学定理，其是

可靠的且足够的。（6）讨论 L_1^S 的性质。主要讨论的是 $L-S$ 性和紧性，而没有讨论所谓的可插入性。从模型论的角度看，这二性质是 L_1^S 的非常好的性质，其使得 L_1^S 的公式集（协调的）可以有各种各样、十分丰富的模型。但也正是由于这些性质，说明 L_1^S 的表示能力有一定的局限。

现在小结一下 L_2^S 和 L_Q^S。实际上，这两种逻辑演算也是可按 L_1^S，分成六部分进行讨论。只是由于是在讨论 L_1^S 基础上作的研究，为避免重复才讨论得较少。L_2^S 不同于 L_1^S 的最根本处，是引入了二阶变元和二阶量词。正由于此，克服了 L_1^S 的某些局限性。但因为 L_2^S 的二阶公式蕴涵了不可数无穷，而其推理规则又是形式的，造成了 L_2^S 是不完全的逻辑系统；但这并不排除增加非形式的推理规则，使其达到完全。L_Q^S不同于 L_1^S 的最根本处，是引入“有不可数无穷多 x，使……”量词。这种逻辑克服了 L_1^S 不能将可数无穷从不可数无穷中分出来的缺点，但对更高的不可数无穷，仍难以界清。从这个意义说，L_Q^S没有比 L_1^S 强多少。

现在讨论有穷性逻辑 $L_{\neg}^S$。其是上述 L_1^S，L_2^S，L_Q^S等等逻辑的语法抽象。从对 $L_{\neg}^S$ 的研究可见，任何满足有穷性原则(即公式集合、公理集合、推理规则均递归，从而定理集合递归可枚举）的逻辑，均不能刻划（或抓住）无穷论域；而如果使这种逻辑描述的数学完全了，也仍不是说，就抓住了无穷论域；只不过是由于所给的非逻辑符号太贫乏，无能力表示这不同的无穷论域，亦即无公式来区分这不同的无穷论域（实际上，正是这种公式造成的不完全性)。不过，我们对 $L_{\neg}^S$ 没有对其语义作讨论，而仅仅是讨论了语法，引入了推理符号“$\vdash$”，没有引入语义推演符号“$\models$”。由于 L_1^S，L_Q^S具有正确性和完全性，而 L_2^S 仅具有正确性，故作为它们的共同性的抽象的 $L_{\neg}^S$，即使引入“$\models$”，讨论它的语义时，也无法使 $L_{\neg}^S$ 的语法推理与语义推演相等价。而只能分情况研究之。有兴趣的读者不妨试研究一下。

现在看一下 $L_{\omega_1\omega}$。它是一种无穷逻辑。其有关量词的规定与 L_1^S 的完全相同。但由于其允许使用无穷合取“$\bigwedge$”和无穷析取

"∨"（严格说，仅允许是可数的无穷合取和析取），故其连接的公式不是简单的有穷组合（顺便说一句，二阶量词公式"$\forall X X a$"尽管形式上是有穷性的，但当进行语义解释时，这种公式却相当于把具有幂集个公式由"∧"连接起来了）。$L_{\omega_1\omega}$的推理规则不全是形式的。也正由于此，$L_{\omega_1\omega}$是完全的逻辑，而L_1^S却不是。不过，很明显，$L_{\omega_1\omega}$又不能描述所有的二阶公式。可以说，现有的数学几乎均可由L_1^S和$L_{\omega_1\omega}$来刻划。因而，若能建立一个包括L_1^S和$L_{\omega_1\omega}$在内的、且既正确又完全的逻辑系统，则无疑是一件有十分重大意义的研究成果。

最后，我们来看一下一般逻辑$\tilde{\mathscr{L}}$和一般数学$\tilde{M}(\tilde{\mathscr{L}})$。它们是一种新的不同于上述种种的逻辑。尽管其也引入了语义推演符号"⊨，及一般推理符号"⊩"，但从实质上讲，该"⊩"同其语义推演符号"⊨"，这从**定理 26** 的证明及从**例 1** 到**例 7** 的证明中是可见的，也就是说，那里仍是在进行"语义验证"。另外，$\tilde{M}(\tilde{\mathscr{L}})$还有几个特点如下：

(1) 其中所给的 X，不应看成是给定的，而应是"任意"给定的，或更确切地说，是集合变元，可取任意给定的 X_0为值；由此，$\tilde{\mathscr{L}}_M$也就成为 X 的函数了，且这种函数的取值，正是命题函数集合，即按语义等价与否将公式分成的等价类。

(2)$\tilde{M}(\tilde{\mathscr{L}})$是可以刻划二阶逻辑中的二阶公式、甚至各阶公式的。这取决于所给的 X 的值 X_0。亦即，若其不是简单的元素集合，而是

$$Y \times P(Y),$$

则既可刻划一阶也可刻划二阶公式，等等。

(3)$\tilde{M}(\tilde{\mathscr{L}})$的一般推理并非刻划的真正的证明过程。实际上，仍是语义验证。因而，也就不会有如前述各种逻辑的所谓完全性定理。

总之，$\tilde{M}(\tilde{\mathscr{L}})$有不少新的特点，但仍需进行深入的研究。

以上这些所谓的"总结"、"小结"、"讨论"等，可能并无反

映了前边所讲的内容，甚至可能有不确切或者错误处。因为这不是以数学方式描述的，所以也就难以说得一清二楚，甚至可能造成错误的理解。总之，若遇到与前边讲的不一致之处，自然以前边的严格的数学描述和证明为准。这里仅仅是给读者提供一个咀嚼和进一步思考的线索。特此说明之。

第五章　集　合　论

本章要讨论朴素集合论、公理集合论和 ZF 系统。这里的朴素集合论主要是由 G、Cantor 所建立的。鉴于有关读者在其他一些课中已有所了解，所以为了节省篇幅，这部分内容，主要是讲解序数。这里所讲的公理集合论，是为避免悖论而发展的，应特别指出的是，这种公理集合论既非实质公理（即对象是先于公理而存在），也非形式公理（即一阶逻辑公理）。故不同于 ZF。本章要讲的 ZF 系统，则是用一阶逻辑表述的公理集合论。如所周知，其也称作 Zermelo-Fraenkel 集合论。

第一节　朴素集合论

关于集合的定义，我们采取肖文灿在1939年的书“集合论初步”中所给出的。他说，集合是“吾人直观或思维之对象，如为相异而确定之物，其总括之全体即谓之集合；其组成此集合之物谓之集合之元素。”“所谓相异者，取二物于此，其为同一，其为相异，可得而决定。而集合所含之元素乃有彼此不同之意味。所谓确定者，此物是否属于此集合，一望而知，至少在概念上可以断定其是否为该集合之元素。盖合于某条件之集合，须其界限分明，不容有模糊不清之弊。”

上述定义是十分清楚明白的。本节就在这种定义基础上讨论基数、序数和超穷归纳法。

1.1　集合之势（基数）

由集合之定义，两个集合**相等**与否，完全由它们的元素相同与否来决定。这无需多加讨论。但要比较集合之大小，却没有这么

简单了。设自然数集合

$$N=\{1,2,3,4,\cdots\},$$

对于 N 的任给的两个有穷子集，那也容易知道它们的大小。但对偶数集 $E=\{2,4,6\cdots\}$，与 N 相比较，当然 $E\subsetneq N$；然而若将1与2对应，2与4对应，3与6对应，…一般地，$n\in N$，与 $2n\in E$ 对应，可见，从对应角度看，N 和 E 的元素一样多，也可以说是一样大小的。不难知道，对有穷集来讲，真子集即比原集合小；而对无穷集来讲，却是不一定的。

称集合 A 与集合 B **对等**，如果在 A、B 间有一个双射（即1-1的满射）。用"$\sim$"记"**对等**"。

易知，对等是集合间的等价关系，故将集合分成了等价类。对任给集合 A，我们定义 A 的势为 $\overline{\overline{A}}=\{B|B\sim A\}$；不难理解，所有与 A 对等的集作为元素而构成之集，它们的特征是彼此对等，对 $\overline{\overline{A}}$ 可理解成是 A 的元素"个数"之抽象。故我们也称 $\overline{\overline{A}}$ 为**基数**，记作 α。

例如，设 $N_n=\{m|m\in N,且\ m\leqslant n\}$，对任 $n\in N$。那么，$\overline{\overline{N}}_n=\{B|B\sim N_n\}$。可知，$\overline{\overline{N}}_n$ 的元素之元素的个数均为 n，比如 n 个人，n 个桌子，n 支笔，等等。所以不妨说，$\overline{\overline{N}}_n$ 即为"个数 n"的抽象。

现在，我们定义集合的势或基数的大小。若基数 $\alpha=\overline{\overline{A}}$，$\beta=\overline{\overline{B}}$，且 A 有子集 A_1，使 $A_1\sim B$，则定义 $\alpha\geqslant\beta$ 或 $\beta\leqslant\alpha$，而又当 $A\not\sim B$（A 不对等于 B）时，定义 $\alpha>\beta$ 或 $\beta<\alpha$。

很易知道，会有 $\alpha\leqslant\beta$ 且 $\beta\leqslant\alpha$ 同时出现的情况。再由上述定义，若此时不能证明 $\alpha=\beta$（即 $A\sim B$），则必有 $\alpha<\beta$ 且 $\beta<\alpha$ 同时出现。故这种比较基数大小的定义趋于失败。因此，必须证明：

定理（Cantor-Bernstein） 若 $\alpha\leqslant\beta$ 且 $\beta\leqslant\alpha$，则 $\alpha=\beta$。

为证上述定理，先证下述引理：

引理 设集合 A_0,A_1,A_2 满足 $A_0\supset A_1\supset A_2$，且 $A_2\sim A_0$，则 $A_1\sim A_0$。

证明　∵$A_2 \sim A_0$，且 $A_0 \supset A_1$，故使 A_2 与 A_0 对等的双射，可使有 $A_3 \subset A_2$，$A_3 \sim A_1$，这样，从 $A_3 \sim A_1$，$A_1 \supset A_2$，类似地有 $A_4 \subset A_3$，$A_4 \sim A_2$，依此类推，得：

$$A_0 \supset A_1 \supset A_2 \supset A_3 \supset A_4 \supset A_5 \supset \cdots$$

$$A_0 \sim A_2, A_1 \sim A_3, A_2 \sim A_4, A_3 \sim A_5, \cdots$$

∵上述对等均是同一双射，

∴　$A_0 - A_1 \sim A_2 - A_3, A_1 - A_2 \sim A_3 - A_4, A_2 - A_3 \sim A_4 - A_5, \cdots$

令 $D = A_0 \cap (\bigcap_{i=1}^{\infty} A_i)$，则有：

$$A_0 = (A_0 - A_1) \cup (A_1 - A_2) \cup (A_2 - A_3) \cup (A_3 - A_4) \cup \cdots \cup D$$

$$A_1 = (A_1 - A_2) \cup (A_2 - A_3) \cup (A_3 - A_4) \cup (A_4 - A_5) \cup \cdots \cup D$$

上述两式中各项均互不相交，且如双箭头所指的双射，可将 A_0 和 A_1 对等起来。故 $A_0 \sim A_1$。

现在，使用该引理，来证 **Cantor-Bernstein 定理**：由 $\alpha \leqslant \beta$，$\beta \leqslant \alpha$，即知 $A \sim B_1 \subset B$，$B \sim A_1 \subset A$，从 $B_1 \subset B \sim A_1$，故有 $A_2 \subset A_1$，使 $A_2 \sim B_1$，因而，$A \supset A_1 \supset A_2$，且 $A \sim A_2$，使用引理，可得 $A \sim A_1 \sim B$，故证毕。

从 **Cantor-Bernstein 定理**，可知：$\alpha = \beta$，$\alpha < \beta$ 或 $\alpha > \beta$ 三者中至多有一个成立（∵由定义，显然 $\alpha = \beta$ 和 $\alpha < \beta$ 不能同时成立，而同样，$\alpha = \beta$，$\alpha > \beta$ 也不能同时成立；而若 $\alpha < \beta$，$\alpha > \beta$ 同时成立，由定义，即意味着 $\alpha \leqslant \beta$，$\beta \leqslant \alpha$，$\alpha \neq \beta$ 同时成立，但从定理，$\alpha \leqslant \beta$ 且 $\beta \leqslant \alpha$ 即得 $\alpha = \beta$，矛盾）。

但若要上述比较基数大小的定义无问题，还必须使 $\alpha = \beta$，$\alpha < \beta$ 或 $\beta < \alpha$ 中至少有一个成立。而要证明这一点，则要作如下很长的介绍。

称凡与 N 对等的集合为**可数集合**。

例 1　若用 $\aleph_0$，记 $\overline{\overline{N}}$，用 C 记 $\overline{\overline{R}}$，R 为实数集，如所周知：$C > \aleph_0$[1)]，且 $C = P(N)$[2)]，$P(N)$ 为 N 的幂集。那么，至今尚无结论的 Cantor 猜想即为 CH：

不存在 μ，使 $C > \mu > \aleph_0$。

但是，如后面所指出，现在不少人认为：Cantor 猜想不成立。

1.2 有序集

设给定集合 A。若在 A 的元素间可建立一个二元关系 $<$（不妨称作"先于"），使得：

(1) 对 A 的任二不同的元素 a，b，若 $a<b$，则不能有 $b<a$（或 $b\not<a$）；

(2) 对 A 的任三个不同元素 a，b，c，若 $a<b$，$b<c$，则 $a<c$；

那么，称 A 为**全序集**，$<$ 为 A 上的**全序关系**。

设 $<$ 为全序集 A 上的全序关系。称 $a_0\in A$ 为 A 的**首元素**（或**最小元素**），如果 A 的任何其他元素 a，均有 $a_0<a$；称 $a_f\in A$ 为

---

1)令 $f(x)=\mathrm{tg}\,\frac{2x-1}{2}\pi$；

显然它是从 (0，1) 到 R 的双射，从而 $\overline{\overline{(0,1)}}=\overline{\overline{R}}$。

现在利用 (0，1) 的不可数性来说明 R 不可数。

对于任意给定的从 N 到 (0，1) 的一对一映射 g，设：

$g(1)=0_1x_{11}x_{12}x_{13}\cdots$，$g(2)=0_1x_{21}x_{22}x_{23}\cdots$，$g(3)=0_1x_{31}x_{32}x_{33}\cdots$，

构造 $x=0_1x_1x_2x_3\cdots$，其中 $x_k\neq 0,9,x_{kk}(k=1,2,\cdots)$，

那么 $x\in(0,1)$ 且对任意自然数 n，$g(n)\neq x$。

所以 g 不可能是满射，故 (0，1) 不可数，从而 R 也不可数。

由此获得 $C>\aleph_0$。

2)显然，$P(N)$ 对等于从 N 到 {0，1} 的一切映射所构成的集合。

现定义 G 如下：

对任一从 N 到 {0，1} 的映射 h，命 $G(h)=\sum\limits_{n=1}^{\infty}\frac{h(n)}{2^n}$；

显然，$E_G=[0,1]$，且易知，G 不是一对一的，例如：

$$\frac{1}{2^n}+\frac{1}{2^{n+1}}+\cdots=\frac{1}{2^{n-1}}+\frac{0}{2^n}+\frac{0}{2^{n+1}}+\cdots,$$

但这样的数或是1，或是 $\frac{m}{2^n}(m=1,3,5,\cdots 2^n-1)$，只有可数个，所以，

$$\overline{\overline{P(N)}}=C+\aleph_0=C。$$

A 的**末元素**（或**最大元素**），如果 A 的任何其他元素 a，均有 $a<a_f$。对任 $a\in A$，称 $A_a=\{x|x<a, x\in A\}$ 为 A 中由 a 确定的**截段**。容易知道，$a\notin A_a$，而当 a_0 为 A 的首元素时，$A_{a_0}=\phi$。

称全序集 A 是**稠密的**，如果对其任给二个不同元素 a，b，均会有 $c\in A$，使 $a<c<b$。

对全序集 A 的任二不同元素 a，b，若 $a<b$，且无 $c\in A$，使 $a<c<b$，则称 a 是 b 的**直前元**，而 b 是 a 的**直后元**。

请读者结合下述例 2中给出的不同的全序集，理解上述各概念。

例2 (a) 1，2，3，4，…；

(b) …4，3，2，1；

(c) 1，3，5…2，4，6，…；

(d) …5，3，1，2，4，6，…；

(e) 1，3，5，……6，4，2；

(f) …5，3，1，…6，4，2；

(g) 有理数集 Q 及其上的小于关系<。

1.3 序型

现在，来研究全序集的比较。即除了用“对等”作为尺度，比较元素的“个数”外，还要比较元素之间的次序。如下给出定义。

称全序集 A 和全序集 B 是**序等的**，记为 $A\simeq B$，如果有 A 到 B 的双射 φ，满足：对任 $a_1, a_2\in A$，当 $a_1 \underset{A}{<} a_2$ 时，恒有 $\varphi(a_1)\underset{B}{<}\varphi(a_2)$。注意，这里 $\underset{A}{<}$，$\underset{B}{<}$ 分别是 A，B 中的全序关系，而 $\varphi(a_1)$，$\varphi(a_2)\in B$。

显然有：若 $A\simeq B$，则 $A\sim B$；而当 A，B 有穷时，其逆也是成立的（当然，要首先建立 A，B 的全序关系）。上述例 2中的 (a) 和 (b) 显然是对等的，然而，并不序等。

易知，序等是全序集合间的等价关系。故将全序集合分成了等价类。对任给全序集 A，我们定义 A 的**序型**为

$$\overset{\simeq}{A}=\{B|B\simeq A, B \text{ 为全序集}\}。$$

例如，设 N 中的序关系为"小于"，并记 $\overline{\overline{N}}$ 为 ω。上述**例 2** 中的（b）若记为 N^*，则 $\overline{\overline{N^*}}\neq\omega$。设 I 为整数集，其上的序关系为"小于"，并记 $\overline{\overline{I}}$ 为 π。而对任 $n\in N$，$\overline{\overline{N_n}}=\overline{\overline{N_n}}$，记作 n。

设任给全序集 A，并令 $H(A)=\{A_a|a\in A\}$，且定义 $H(A)$ 上的序关系 $\underset{H(A)}{<}$ 为：任 $A_a, A_{a'}\in H(A)$，$A_a<_{H(A)}A_{a'}$ 当且仅当 $A_a\subset A_{a'}$（亦即 $a\underset{A}{<}a'$）。

不难知道，对任可数的全序集 A，可在有理数集 Q 中找到全序子集 Q_0，使 $A\simeq Q_0$，留作练习。

1.4　序型的运算

设全序集 A，B，且 $A\cap B=\phi$，而 $\overline{\overline{A}}=\mu$，$\overline{\overline{B}}=\gamma$；构造全序集 $S=A+B$，S 的元素为 $A\cup B$ 中的元素，且 S 中的序关系按 A，B 中的序关系，而当 $a\in A$，$b\in B$ 时，总令 $a<b$。称 S 为 A 和 B 的**全序和**. 又设 $\overline{\overline{S}}=\xi$，则定义 $\mu+\gamma=\xi$。

由上述定义，易知：$1+\omega=\omega$，但 $\omega+1\neq\omega$。可见，序数加法不满足交换律。事实上，$\omega+1$，$\omega+2$，$\omega+3$，…，$\omega+n$，…这些均相互不相等。

若定义 $\omega+\omega$ 为 $\omega\cdot 2$，则 $(\omega+\omega)+1$，$(\omega+\omega)+2$，…即为 $\omega\cdot 2+1$，$\omega\cdot 2+2$，…；类似地，定义 $\omega\cdot 2+\omega$ 为 $\omega\cdot 3$，定义 $\omega\cdot 3+\omega$ 为 $\omega\cdot 4$，…。

现在推广上述全序和的定义。设 D 是全序集（作为指标集），对每个 $\lambda\in D$，有全序集 A_λ，且当 $\lambda\neq\lambda'(\lambda,\lambda'\in D)$ 时，$A_\lambda\cap A_{\lambda'}=\phi$。构造全序集 $S=\sum_{\lambda\in D}A_\lambda$，$S$ 中元素为 $\bigcup_{\lambda\in D}A_\lambda$ 中的元素，且 S 中的序关系按各 A_λ 的序关系，而当 $a\in A_\lambda$，$a'\in A_{\lambda'}$，且 $\lambda\underset{D}{<}\lambda'$ 时，总令 $a<a'$。称 S 为 $\{A_\lambda|\lambda\in D\}$ **的全序和**。又设 $\overline{\overline{S}}=\xi$，则定义 $\sum_{\lambda\in D}\overline{\overline{A_\lambda}}=\xi$。

由上述推广的全序和的定义，有：

$3+3+3+\cdots=\sum_{n\in N}3$，记为 $3\cdot\omega$；易知 $3\cdot\omega=\omega\neq\omega\cdot 3$。类似地，

$\omega+\omega+\omega+\cdots=\sum_{n\in N}\omega$，记为 $\omega\cdot\omega$，又记为 ω^2，而 $\omega^2+\omega^2+\omega^2+\cdots=\sum_{n\in N}\omega^2$，记为 $\omega^2\cdot\omega$，又记为 ω^3，… 。

1.5 良序集

若全序集 A 的任一非空子集必有首元素，则这种 A 称作**良序集**。上述例2中的（a）和（c）为良序集，而（b），（d），（e），（f）和（g）均为非良序集。

设 A 和 B 为良序集，则有下述性质。

性质1 A 的子集均为良序集。

性质2 与集合 A 序等的集合也是良序集。

证明 注意到只能首元素与首元素互为双射的象。易证明结论。兹略去。

性质3 A 中除末元素外，任何元素均有直后元。

证明 任取 $a\in A$，若 a 不是末元素，则作 $A_a\cup\{a\}$，$A-(A_a\cup\{a\})$是 A 的非空子集，其有首元素。那么，该首元素必是 a 的直后元。

性质4 A 中不存在元素的序列$\{a_n\}_{n\in N}$，且满足$\cdots<a_3<a_2<a_1$。

性质5 $A'\subset A$，则使 A 序等于 A' 的双射 φ，必然对任 $a\in A$，$\varphi(a)\geqslant a$(即 $\varphi(a)=a$ 或 $\varphi(a)>a$)。

证明 用反证法。若该性质不成立，则必有 $a\in A$，使 $\varphi(a)<a$，那么，令

$$M=\{a\mid\varphi(a)<a,a\in A\},$$

则 $M\neq\phi$，且 M 也为良序集（由性质1），故 M 必有首元素 a_0，当然也有 $\varphi(a_0)<a_0$；又∵$a_0,\varphi(a_0)\in A$，由 φ 的保序性，得 $\varphi(\varphi(a_0))<\varphi(a_0)$，而从 M 定义可见，$\varphi(a_0)\in M$，这与 a_0为 M 的首元素相矛盾。因而证毕。

性质6 A 不能与其任一截段序等，也不能与其子集的任一截段序等。A 还不能与其任一截段的子集序等。

证明　**由性质5易证，略。**

性质7　A 的任二不同截段不能序等。

证明　**由性质5易证，略。**

性质8　对 A，B 言，或者 $A \simeq B$，或者其中之一序等于另一的截段。

证明　首先，用反证法，从性质7易证：任二序等的良序集，其序等的双射是唯一的（*）。下面证明中要使用这一结果（留作练习）。

其次，令 $M_A = \{a | a \in A$，且 A_a 与某 B_b 序等，$b \in B\}$，易知，A 的首元素属于 M_A，且如果某 $a_1 \in M_A$，则任 $a_2 \in A_{a_1}$，该 $a_2 \in M_A$（留作练习）。那么，或者 $M_A = A$，或者有 $a \in A$，$M_A = A_a$。同样，令 $M_B = \{b | b \in B$，且 B_b 与某 Aa 序等，$a \in A\}$。那么，或者 $M_B = B$，或者有 $b \in B$，$M_B = B_b$。

第三，我们来证 $M_A \simeq M_B$：对于任 $a \in M_A$，由 M_B，M_A 的构造，必有某 $b \in M_B$，使 $A_a \simeq B_b$，从本定理证明开始的结果（*），知这种 b 是唯一的；反之，对于 $b \in M_B$，亦有且仅有一个 $a \in M_A$，使 $B_b \simeq A_a$；故这样由 a 到 b 构造的从 M_A 到 M_B 的对应 φ 是双射。现在来证这 φ 还是保序的。设 $a_1, a_2 \in M_A$，且 $a_1 < a_2$，$\varphi(a_1) = b_1$，$\varphi(a_2) = b_2$，则由 $A_{a_2} \simeq B_{b_2}$，其中，$A_{a_1} = (A_{a_2})_{a_1}$ 序等的部分设为 $(B_{b_2})_{b_0}$，再由 B 中只有一个截段与 A_{a_1} 序等，故只能是 $b_0 = b_1$，即得 $b_1 < b_2$。故 $M_A \simeq M_B$。

总括上述情况，得下述四种可能：(1) $M_A = A$，$M_B = B$；(2) $M_A = A$，$M_B = B_b$，$b \in B$；(3) $M_A = A_a$，$a \in A$，$M_B = B$；(4) $M_A = A_a$，$M_B = B_b$，$a \in A$，$b \in B$。但其中第(4)种可能是不会发生的。因为，不然，由于有 $A_a = M_A \simeq M_B = B_b$，故从 M_A 的定义知，$a \in M_A$，这又与 $M_A = A_a$ 矛盾（$\because a \in A_a$）。而其他三种可能即是该性质所需证明的。

由上述各性质，即可在良序集间建立全序关系“$<$”：

1. 任良序集 A 均大于其任一截段 A_a，$a \in A$；
2. 任良序集 A 的二截段 A_a，A_b，若 $a < b$，则 $A_a < A_b$；

3. 任良序集 A，B，知其或者 $A=B$，或者 $A<B$，或者 $B<A$；

4. 设 a 为 A 的首元素，则 $A_a=\phi$，其为最小的良序集。

性质 9 设 $\mathfrak{M}$ 是互不序等的良序集为元素的集族，则 $\mathfrak{M}$ 中有最小的良序集。

证明 设 $A\in\mathfrak{M}$，且设 A 不是最小的（否则，A 即所求），则有较 A 小的 B，是由**性质8**，必有 $a\in A$，使 $B\simeq A_a$；令

$$A_0=\{a|a\in A,\text{且有 } B\in\mathfrak{M},\text{使 } B\simeq A_a\},$$

故 $A_0\subset A$，由**性质1**，A_0也良序，再从有较 A 小的 B，故从 A_0定义知 $A_0\neq\phi$，因而 A_0有最小元素（首元素），设为 a_0，再从 A_0的定义，有 $B'\in\mathfrak{M}$，使 $B'\simeq A_{a_0}$。（请注意，A_{a_0}是 A 的由 a_0确定的截段，而非 $(A_0)_{a_0}$）。那么，显然，该 B'即为 $\mathfrak{M}$ 中最小的良序集。

1.6 序数

称良序集的序型为**序数**。无限的良序集的序型称作**超限数**（也常称作超穷数）。

例 3 0，1，2，3，$\cdots$；

ω，$\omega+1$，$\omega+2$，$\cdots$；

$\omega+\omega$ 即 $\omega\cdot2$，$\omega\cdot2+1$，$\cdots$；

$\omega\cdot2+\omega$ 即 $\omega\cdot3$，$\omega\cdot3+1$，$\cdots$；

……

$$(\cdots(\cdots(\underbrace{(\omega+\omega)+\omega)+\cdots+\omega)+\cdots)}_{\sum\limits_{n\in N}\omega}$$

即 $\omega\cdot\omega$ 即 ω^2，ω^2+1，$\cdots$；

$\omega^2+\omega$，$(\omega^2+\omega)+1$，$(\omega^2+\omega)+2$，$\cdots$；

$(\omega^2+\omega)+\omega$ 即 $\omega^2+\omega\cdot2$，$(\omega^2+\omega\cdot2)+1$，$\cdots$；

$((\omega^2+\omega)+\omega)+\omega$ 即 $\omega^2+\omega\cdot3$，$(\omega^2+\omega\cdot3)+1$，$\cdots$；

……

$$(\cdots(\cdots((\omega^2+\underbrace{\omega)+\omega)+\cdots+\omega)+\cdots)}_{\omega^2+\sum\limits_{n\in N}\omega}$$

即 $\omega^2+\omega\cdot\omega$ 即 $\omega^2+\omega^2$，$\omega^2+\omega^2+1$，…；

…………

$\underbrace{\omega^2+\omega^2+\cdots}_{\sum_{n\in N}\omega^2}$ 即 $\omega^2\cdot\omega$ 即 ω^3，ω^3+1，…；

…………

$$(\cdots\ (\cdots\ \underbrace{(\omega\cdot\omega)\ \cdot\omega\cdot\ \cdots\ \cdot\omega)\ \cdot}_{\prod_{n\in N}\omega}\ \cdots)$$

即 ω^ω，$\omega^\omega+1$，$\omega^\omega+2$，…。

…………

上述这些均为序数，且非自然数表示的序数均为超限数（当然，超限数远不止上述这些，将在1.7和1.8中讨论）。

序数的大小由相应的良序集的大小来确定，这样，任何序数均可进行大小比较。

由于良序集的性质，序数也有下述性质。

性质 1 序数集合中必有最小的序数。

性质 2 序数的集合中，以序数大小为序，则构成良序集。

性质 3 令 $W_\mu=\{\gamma|\gamma<\mu,\ \gamma$ 为序数$\}$，μ 为序数，则 W_μ 也为良序集，且 $\overset{\simeq}{W}_\mu=\mu$。

证明 首先，取 A 为良序集，且 $\overset{\simeq}{A}=\mu$，令 $H(A)=\{A_a|a\in A\}$，由本节1.3易知：$H(A)\simeq A$。

其次，证明 $H(A)\simeq W_\mu$：令 $\Psi(A_a)=\overset{\simeq}{A}_a$；因为任 $\overset{\simeq}{A}_a<\overset{\simeq}{A}=\mu$，故 Ψ 是从 $H(A)$ 到 W_μ 的；又因为当 $A_a\neq A_b$ 时，也有 $\overset{\simeq}{A}_a\neq\overset{\simeq}{A}_b$，故 Ψ 又是单射（1-1的）；再因为 W_μ 中的每个序数 $\gamma<\mu$，而 $\overset{\simeq}{A}=\mu$，故必有 $a\in A$，使 $\gamma=\overset{\simeq}{A}_a$，故 Ψ 又是满射；再注意到 $H(A)$ 中的序关系为：$A_{a'}\underset{H(A)}{<}A_a$ 当且仅当 $A_{a'}\subset A_a$（亦即 $a'\underset{A}{<}a$），故 Ψ 即为使 $H(A)\simeq W_\mu$ 的序等双射。

因而，由上述得：$W_\mu\simeq H(A)\simeq A$，而 $\overset{\simeq}{A}=\mu$，故 $\overset{\simeq}{W}_\mu=\mu$。

性质 4 设 A 为良序集，且 $\overline{\overline{A}}=\mu$，则 A 中的元素可用小于 μ 的序数来编号，即 $A=\{a_\gamma|\gamma<\mu\}$。

证明 由**性质 3** 的证明中立得。

性质 5 S 是序数的集合，则有一序数，其大于 S 中任一序数。

证明 首先，对任何序数 μ，必有大于 μ 的序数 $\mu+1$；因此，若 S 中有最大序数时，则**性质 5** 显然成立。故只要证：当 S 中无最大序数时，定理也成立。如下证明之。因为 S 是序数集，故按序数大小构成全序集，对任 $\mu\in S$，取良序集 A_μ，使 $\overline{\overline{A}}_\mu=\mu$，自然可使得对 $\mu_1,\mu_2\in S$，当 $\mu_1\neq\mu_2$ 时，$A_{\mu_1}\cap A_{\mu_2}=\phi$；故可作 $\{A_\mu|\mu\in S\}$ 的全序和 $\mathbb{S}=\sum\limits_{\mu\in S}A_\mu$，又设 $\overline{\overline{\mathbb{S}}}=\xi$，则 $\xi=\sum\limits_{\lambda\in S}\overline{\overline{A}}_\lambda=\sum\limits_{\mu\in S}\mu$。现在证 $\xi>\mu$ $(\in S)$：因为 S 中无最大的序数，故对任 $\mu\in S$，均有 $\mu'>\mu$，$\mu'\in S$，设 b 是 $A_{\mu'}$ 的首元素，那么，A_μ 必是 $\mathbb{S}$ 的某截段 $\mathbb{S}_b$ 的子集，故由良序集的**性质 6**，$\mathbb{S}$ 不会序等于 A_μ，故 $\xi>\mu$；证毕。

性质 6 序数 $\mu+1$ 称为序数 μ 的**直后序数**，而 μ 称为 $\mu+1$ 的**直前序数**。请读者证明：任序数 μ 均有直后序数，但不一定有直前序数。

我们用 **0** 记良序集 ϕ 的序型，并称有直前序数的序数为**后继序数**，而称无直前序数的序数为**极限序数**。这样，序数可分成0，后继序数和极限序数三种。

1.7 可数的超限数

称可数的良序集的序型为**可数超限数**。用 Z_1 表示所有的可数超限数之集，则 Z_1 也是良序集。**例 3** 所一一列出的即所有的可数超限数。

本小节和下一小节，我们要构造出不可数序数和不可数基数。

命题 1 ω 是最小的可数超限数。

证明 易证。留作练习。

命题 2 若 $\mu\in Z_1$，则 $\mu+1\in Z_1$。

证明　易证。留作练习，

命题 3　设可数的序数集 $S \subset Z_1$，又设 γ 是大于 S 中所有序数的最小序数（由序数的**性质 5**，可以这样假设），则 $\gamma \in Z_1$。

证明　若 S 中有最大的序数 ξ，则 $\gamma = \xi + 1$ 属于 Z_1（由命题 2），而 γ 又是大于 S 中所有序数的最小序数，故命题成立。而当 S 中没有最大的序数时，且由 γ，作 W_γ（如序数的**性质 3**中所作），那么有 $W_\gamma = \bigcup\limits_{\mu \in S} W_\mu$，证明如下：

若 $\xi \in \bigcup\limits_{\mu \in S} W_\mu$，则必有 $\mu_0 \in S$，使 $\xi \in W_{\mu_0}$，故由 W_{μ_0} 的定义，知 $\xi < \mu_0$，而 $\mu_0 < \gamma$，故 $\xi < \gamma$，$\therefore \xi \in W_\gamma$；

若 $\xi \in W_\gamma$，则 $\xi < \gamma$；而 $\because \gamma$ 是大于 S 中序数的最小的序数，$\therefore$ ξ 不能大于 S 中所有的序数，故不妨设有 $\mu_0 \in S$，$\mu_0 \geqslant \xi$；又 $\because S$ 中没有最大的序数，故有 $\mu' \in S$，$\mu' > \mu_0$；那么，由 $W_{\mu'}$ 的定义，$\xi \in W_{\mu'}$（$\because \mu' > \mu_0 \geqslant \xi$），因而 $\xi \in \bigcup\limits_{\mu \in S} W_\mu$。

由于 Z_1 是所有的可数超限数之集合，故任 $\mu \in S$，W_μ 均可数（$\because \overset{\approx}{W}_\mu = \mu \in S \subset Z_1$），再由 S 可数，故 $\bigcup\limits_{\mu \in S} W_\mu$ 也可数（可数个可数集之并，也可数）；那么，由上所证 $W_\gamma = \bigcup\limits_{\mu \in S} W_\mu$，故 W_γ 也可数，因而 γ 也可数（$\because \gamma = \overset{\approx}{W}_\gamma$），$\therefore \gamma \in Z_1$。证毕。

命题 4　Z_1 不可数，并记 $\overline{\overline{Z_1}}$ 为 $\aleph_1$。

证明　由**命题 3**，若 Z_1 可数，则比 Z_1 中所有的序数均大的最小序数 μ 也应属于 Z_1，故 μ 也可数，因而得 $\mu > \mu$，矛盾。证毕。

命题 5　比 Z_1 中任何序数均大的最小序数记为 W_1，当然，W_1 是最小的（或说是第1个）不可数超限数。

证明　显然。事实上，由序数的**性质 3**，$W_1 = \overset{\approx}{N} \cup \overset{\approx}{Z}_1 = \overset{\approx}{N} + \overset{\approx}{Z}_1 = \omega + \overset{\approx}{Z}_1$，而 $N = \{0, 1, 2, \cdots\}$。

命题 6　$\aleph_1$ 是最小的不可数基数。

证明　由 $\omega_1 = \omega + \overset{\approx}{Z}_1$，$\omega_1$ 是最小的不可数超限数，故 $\aleph_1$ 是最小的不可数基数。

设 μ 为序数，A 为良序集，且 $\tilde{\tilde{A}}=\mu$，而 $\varphi(\mu)=\overline{\overline{A}}$，称 $\varphi(\mu)$ 是序数 μ 的基数。

那么，对任何 $\mu\in Z_1$，$\varphi(\mu)=\aleph_0$

对任何基数 $\aleph$，令 $Z(\aleph)=\{\mu|\varphi(\mu)=\aleph\}$，则 $Z(\aleph)$ 也为良序集，故有最小元 $\mu_\aleph$，使 $\varphi(\mu_\aleph)=\aleph$。当 $\aleph$ 为超限的基数时，则相应的 $\mu_\aleph$ 必为极限序数（因为，否则，其直前序数 $\mu'_\aleph$ 也会满足 $\varphi(\mu'_\aleph)=\aleph$，而 $\varphi(\mu'_\aleph+1)$ 即 $\varphi(\mu_\aleph)=\aleph$，与 $\mu_\aleph$ 的“最小”假设相矛盾）。称这种序数为初始序数。如对 $\aleph_0$ 而言，其相应的初始序数为 ω（也记为 ω_0），而对 $\aleph_1$ 言，其相应的初始序数为 ω_1。

显然，$Z(\aleph_0)$ 即上述 Z_1。

不难将上述对 $Z(\aleph_0)$ 即 Z_1 的从**命题 2** 至**命题 6** 的证明，移至对 $Z(\aleph_1)$ 的有关证明，从而得到第二个不可数超限数 ω_2 和第二个不可数基数 $\aleph_2$。且将这种证明推广至对任 $Z(\aleph_\iota)$，即可得到下述两个序列：

$\omega_1,\omega_2,\omega_3,\cdots$ 和 $\aleph_1,\aleph_2,\aleph_3,\cdots$。

更可得到下述结果：对每个序数 ρ，恒有一相应的基数 $\aleph_\rho$，使此对应是1－1且保序的。即所有的超穷的基数可按大小排成：

$\aleph_0,\aleph_1,\aleph_2,\cdots,\aleph_\omega,\aleph_{\omega+1},\cdots,\aleph_{\omega_1},\aleph_{\omega_1+1},\cdots$

在作了上述讨论后，连续统猜想可表述为

$$C=\aleph_1(\text{或 } 2^{\aleph_0}=\aleph_1)。$$

现在，对本小节的研究作一下小结。(1) 从有穷到无穷采用的方法是将“所有的”有限数作成一个集合，且按从小到大的次序排好，得到 $\omega=\tilde{\tilde{N}}$，而 $N=\{0,1,2,\cdots\}$，可取该等价类的代表为 N；(2) 从可数到第一个不可数有类似的方法，$Z_1=\{\mu|\varphi(\mu)=\aleph_0\}$，即“所有的”基数为 $\aleph_0$ 的序数“之和”；(3) 从可数到不可数还有另一种方法，即“所有的”$\aleph_0$ 的子集“之和”，$P(\aleph_0)=2^{\aleph_0}$；(4) 而连续统猜想 CH 即是说，由这两种方法得到的“不可数”是

相同的。

然而，Cohen 在其经典的论著末尾却写道："人们终将接受的观点是：CH 显然是错误的。人们所以接受无穷公理，可能是觉得，每次只增加一个集合，就企图穷尽整个全域的观点是荒谬的。接受更高阶的无穷公理，也是基于同样理由。$\aleph_1$是全体可数序数之集，且这种得到$\aleph_1$的方式，仅仅是一种产生更大基数的最简单、最特殊的方法。""而 C 却是由全新的更强有力的原理——幂集公理所产生。由此看来，利用替换公理去构造更大基数就一定会达到 C 的观点，是很难为人们接受的。事实上，C 应比$\aleph_n$，$\aleph_\omega$，$\aleph_\alpha$，($\alpha=\aleph_\omega$)等都大。""C 作为由陡增的新公理所产生的异乎寻常丰富的集合。用渐进构造的方法永远不可能达到。" 构造比$\aleph_0$更大无穷的两种方法，当用于有穷集时，它们的结果甚为不同。这也使我更相信上述观察。

1.9 超限归纳法和良序定理

如所周知，对自然数的性质 $p(n)$，我们可以施以归纳法进行证明，这种归纳法可称作**有穷归纳法**；而对有关序数 μ 的命题，我们也可以施行归纳法进行证明。这种归纳法称作**超限归纳法**。

设 $T(\mu)$是一个有关序数的命题，如果满足

(1) 对某序数 μ_0、$T(\mu_0)$正确；

(2) 若对任 γ，μ ($\mu_0\leqslant\mu<\gamma$)，设 $T(\mu)$正确，能证得 $T(\gamma)$ 也正确；

那么，对任何序数 $\mu\geqslant\mu_0$，$T(\mu)$均正确。

现在，来证明超限归纳法的正确性：用反证法。若有 $\mu>\mu_0$，使 $T(\mu)$不正确，故必有最小的 $\mu'>\mu_0$，使 $T(\mu')$不正确，也就是说，对任何 μ，$\mu_0\leqslant\mu<\mu'$，$T(\mu)$均正确；但由上述 (2)，又得到 $T(\mu')$ 是正确的，矛盾。证毕。

由**1.8** 的讨论，我们不但可以比较各个不同的序数，也可以比较各个不同的基数。但是，任给一个集合，能否将它排序之后，使其变成一个良序集?答案是肯定的。但有一个前提，即必须首先承

认选择公理（AC）。就是说，使用选择公理（AC），即可以证明良序定理。

选择公理（AC）：对若干个非空集合作成的集族 F，均有 F 上的选择函数 f，使对任 $S\in F$，均有 $f(S)\in S$。

良序定理　任何集合 S 均可良序化，即可将其元素排序，使其成为良序集。

证明　由 S，可得其幂集组成的集族 $P(S)$，由 AC，可有 $P(S)-\{\varnothing\}$上的选择函数 f，那么，可将 S 如下排序：

$a_0=f(S)$，即为首元素，

$a_1=f(S-\{a_0\}),a_2=f(S-\{a_0,a_1\}),\cdots\cdots$

使用超限归纳法，假设对任 $\mu<\gamma$，均定义了 a_μ，则定义 $a_\gamma=f(S-\ \{a_\mu|\mu<\gamma,\ a_\mu\in S\})$，

这样，选尽 S 之后，按次序新定义的 S 各元素，即构成良序集。

以上是使用 AC 证得了良序定理。实际上，使用良序定理，也可以证明选择公理，就是说，这二者是相等价的。在方嘉琳的书中，作者列出并证明了 AC 的九种等价形式，张锦文在《公理集合论导引》中列出并证明了八种等价形式，而谢邦杰列出了24种等价形式，并证明其中四种。我们将在第三节"ZF 系统"的3.2中，再列出另外九种形式，并给出其中七种相互等价的证明。

在1.8小节末尾，主要是摘译 Cohen 的话，对 CH 作一小结。这里，也作为介绍朴素集合论的结束，对 AC 作一点汇总和评论。AC 不但是集合论的重要公理，也是其他数学论证的一种基本方法。AC 的种种等价形式，正是适应现代数学的不同领域的需要而产生。但自从使用 AC，证得了诗论式的"**分球定理**"（即任一闭球体 W，可经分切组合而构成两个完全与 W 相同的闭球体）后，AC 的正确性，就成为数学中一个很重要的问题了。在 Jech 的二书中，都介绍了**素理想定理**（即 PIT：每个布尔代数 B 上均有一个素理想）。并知：$AC\Rightarrow PIT\nRightarrow AC$；且有：$PIT\Leftrightarrow$逻辑完全性定理$\Leftrightarrow$逻辑语义紧性定理；还知：$PIT\Rightarrow$布尔代数的 Stone 表示定理。仅用 PIT 而无需借助 AC 即可证得代数和拓朴中的很多结果。由此可

见，研究比 AC 弱的公理是十分重要的课题。关于这些，读者也可参阅张锦文、谢邦杰两书的附录。

第二节　公理集合论（非形式公理系统）

朴素集合论尽管已成果卓著，但却蕴涵着一些矛盾。Cantor 等研究者就已意识到其中一些，但由于没有分析清原因，故只是消极地避开，而没有去设法解决它们。

著名的 Russell 悖论就是最著名的一个。作集合 $T=\{x|x\notin x\}$，那么可导致：$T\in T$ 当且仅当 $T\notin T$。这矛盾，实际上是从"T 是以所有的集合作为元素的集合"所引起。因为，显然不会有 $x\in x$ 的，故任集合 X，均满足 $x\notin x$；可是 T 的定义又要求 $T\in T$。

类似的悖论，可从"所有的序数作为元素的集合"中引起。问题在于，由引号中的定义，结合第一节的结果，这集合也是序数。那么，也陷入了同样的循环：该集合正在构造之中，又要作为元素放入自己之中了。因而，这集合也就无法形成。故不能将这种构造过程得到的对象算作集合。也可更严格地从逻辑上导出矛盾：设 $T=\{\xi|\xi$ 为序数$\}$，其为良序集，故可设 $\widetilde{\overline{T}}=\gamma$，且我们还知 $\widetilde{\overline{W_\gamma}}=\gamma$，而 $W_\gamma=\{\mu|\mu<\gamma\}$。这就是说，$T$ 序等于它的截段 W_γ（即 T_γ）。这与已证结果相矛盾。

本节，就是要给集合论建立公理系统。这种公理系统，不是实质公理系统，因为所要讨论的对象是由公理而形成的，自然就不是先于公理而存在的。另方面，这种公理系统不是由一阶语言描述的，而是用自然语言（或严格的自然语言，即数学语言）描述的。故这种公理系统也不是形式的。具体说来，集合论公理系统就是从两个集合出发（即空集和无穷集 ω），经若干类似运算的公理，将集合之对应物都给出来，此外，还给出若干条集合必须满足的性质作为公理。然而，这里并未限死集合的界限，这也同第三节所讨论的 ZF 系统是一样的。

2.1 集合论公理

空集公理 A_ϕ 有一个空集合，不妨记作$\varnothing$，该集合中无任何元素。

外延公理 A_e 任集合，都是由它的元素完全决定的。亦即，任二集合，它们相等与否，当且仅当它们彼此的元素同与否。

无序对公理 AB 对任给的二集合 x，y，均有另一集合 S，其以 x 和 y 为元素，记作 $S=\{x, y\}$，也称其为 x，y 的无序对集合。

例 1 由 A_ϕ，$\varnothing$为集合，由 A_ϕ，AB 和 A_e，$\{\varnothing\}=\{\varnothing, \varnothing\}$，为集合，而 $\{\varnothing, \{\varnothing\}\}$ 也为集合。

设 a，b 是集合，用 $\langle a, b\rangle$ 来简记集合 $\{\{a\}, \{a, b\}\}$，那么，使用外延公理 A_e，不难证明：对任集合 a，b，u，v，$\langle a, b\rangle=\langle u, v\rangle$ 当且仅当 $a=u$，$b=v$。

幂集公理 AP 对任集合 S，均有另一记为 $P(S)$的集合，其元素为 S 的任何子集。称 $P(S)$为 S 的幂集，记为 $P(S)=\{x|x\subset S\}$ ($x_1\subset x_2$，即 x_1是 x_2的子集)。

并集公理 AU 对任集合 S，均有另一集合$\bigcup S$，其元素为 S 的元素的元素。记为$\bigcup S=\{x|$有 $y, y\in S$，而 $x\in y\}$。

例 2 设 $S_1=\{a,b,c\}$，则 $P(S_1)=\{\phi, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$(共八个元素)，而$\bigcup(P(S_1))=\{a,b,c\}=S_1$。且易知，对任 S_1, S_2，$S_1\bigcup S_2=\bigcup\{S_1, S_2\}$。

分离公理 AS 对任给集合 S 和任给的性质 $p(x)$，均有另一集合 S_1，其元素为满 $p(x)$的 S 中的元素。记为 $S_1=\{x|p(x)$且 $x\in S\}$。

例 3 设 S_2是集合，而 $p_1(x)$为"$x\in S_1$"，则：$S_1\bigcap S_2=\{x|p_1(x)$且 $x\in S_2\}$；而若令 $p_2(x)$为"$x\notin S_1$"，则 $S_2-S_1=\{x|p_2(x)$且 $x\in S_2\}$。

例 4 设集合 S 不空，则总会有 $S_1\in S$。而令 $p(x)$ 为 "$\forall y(y\in S\rightarrow x\in y)$"，那么，定义

$$\bigcap S = \{x \mid p(x) \text{ 且 } x \in S\},$$

易知，其为 S 中所有元素的共同元素组成。应指出的是："S 不空"是不可缺少的。

例 5 设 S_1, S_2 是集合，则 $S_1 \times S_2 = \{\langle x_1, x_2 \rangle \mid x_1 \in S_1, x_2 \in S_2\}$ 也是集合。留作练习。

分离公理 *AS* 是 Zermelo 引入的。该公理是为克服 Russell 悖论而提出来的。在朴素集合论中，可以称 $T = \{x \mid x \notin x\}$ 是集合，而公理集合论中就不行了。这是因为在朴素集合论中，随便一条什么性质 $p(x)$（比如 $x \notin x$）就可以定义集合 $T = \{x \mid p(x)\}$（有时称这为**概括**公理）；而在公理集合论中，则必须对该性质给予限定在某一已知集合 S 中，即 $T = \{x \mid p(x) \text{ 且 } x \in S\}$。这就是不少数学工作者所称的"研究的任何内容都要限定在一个空间中"。

替换公理 *AR* 设 f 是任一函数，S 是任一集合，则有另一集合 $S' = \mathrm{ran}(f \upharpoonright S)$，即将 f 限定在 S 的值域。

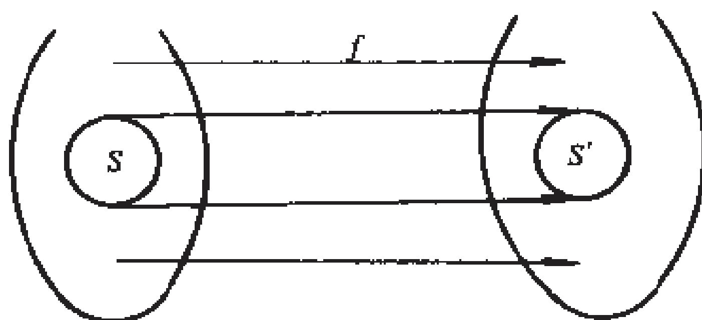


图5-1

可如图 5-1 示 AR。函数 f 是任意的，即其定义域和值域均未限定为集合；但当将定义域限定在集合 S 上时，则所得的值域也为集合。

无穷公理 $A\omega$ 有一集合 $\omega = \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}, \cdots\}$，即其第1个元素为 $\varnothing$，而其后面的每个元素均是以前边所有元素为元素的集合。

正则公理 *AF* 任非空集 S 均有一元素 x，使 $x \cap S = \varnothing$。并称

x 为 S 的极小元。

不难看出，所谓极小元 x，即其不再包含 S 的其他元素。因而可以说，这里的“极小”，是从“包含”意义上而言。

从上可见，非形式的公理集合论，其公理计有九条：

$$\underbrace{A\phi, A\omega}_{\text{直接给出集合}}, \underbrace{AB, AP, AU, AS, AR}_{\text{从已知集合,得新集合}}, \underbrace{Ae, AF}_{\text{限制条件}}。$$

这样，在公理集合论中，集合即是由上述直接或间接得到的具有上述性质的对象，而不再是朴素集合论中，定义含混以至蕴涵矛盾的集合了。正如本节一开始所指出的，公理集合论并未限死集合的界限，即直接或间接地给出一些集合，并给出集合所必须满足的二条性质；至于尚未给出的且又满足该二条件的对象，可以为集合，也可以不为集合。

另外，上述九条公理中，有二条是讲得欠理想的，一是“性质 $p(x)$”（AS 中），二是“函数”（AR 中）。在 ZF 系统中，将这欠理想的二条公理都说清楚了，不再依赖所定义的集合；但是，ZF 却排除了很多应为集合的对象。

为了与这里的公理集合论作对照，我们特提前简单给出 ZF 系统。其是完全用一阶逻辑语言来刻画的。如下：

一阶语言符号集 $S=\{\in\}$，该“$\in$”为一个二元谓词符号。而相应的各条公理为（以相同的符号记之）

A_ϕ: $\exists x\forall y(\neg y\in x)$；

A_ω: $\exists x\forall u(u\in x\leftrightarrow(\forall y(\neg y\in u)\vee\exists u_1(u_1\in x\wedge u=u_1\cup\{u_1\})))$，

其中，“$u=u_1\cup\{u_1\}$“为”$\forall v(v\in u\leftrightarrow v=u_1\vee v\in u_1)$”的简写；

AB: $\forall x\forall y\exists z\ \forall u(u\in z\leftrightarrow u=x\vee u=y)$；

AP: $\forall x\exists y\ \forall u(u\in y\leftrightarrow u\subset x)$，

其中，“$u\subset x$”为“$\forall z(z\in u\rightarrow z\in x)$”的简写；

AU: $\forall x\exists y\ \forall u(u\in y\leftrightarrow\exists z(z\in x\wedge u\in z))$；

AS: $\forall x\exists y\ \forall u(u\in y\leftrightarrow u\in x\wedge P(x))$，

其中，$P(a)$为一阶公式；

AR：$\forall x\exists! yA(x,y)\to\forall x\exists y\forall u(u\in y\leftrightarrow\exists z(z\in x\wedge A(z,u)))$，

其中，$A(a,b)$为一阶公式，而"$\exists! yA(x,y)$"为"$\exists yA(x,y)\wedge\forall y_1,y_2(A(x,y_1)\wedge A(x,y_2)\to y_1\equiv y_2)$"的简写；

Ae：$\forall x,y(\forall z(z\in x\leftrightarrow z\in y)\to x\equiv y)$；

AF：$\forall x(\exists u(u\in x)\to\exists u(u\in x\wedge\forall z(z\in u\to\neg z\in x)))$；

也可在语言符号集 S 中加入常量符号$\varnothing$，即 $S=\{\in, \varnothing\}$，而将 A_ϕ 改为"$\forall u$ $(\neg u\in\varnothing)$"，将 $A\omega$ 改为"$\exists x$ $(\varnothing\in x\wedge\forall y$ $(y\in x\to y\cup\{y\}\in x))$"。试将其中的"$y\cup\{y\}\in x$"改为一阶语言描述的形式。留作练习。

现在对比一下这两个公理系统。

(1) 首先，如在第三章所述，ZF 的非逻辑符号只有"$\in$"，其不过是一个由 ZF 公理所限定的二元关系符号，为了与前一公理中的"$\in$"相区别，在 ZF 中将"$\in$"写成重体的。而当把 ZF 中的"$\in$"作语义解释时，如将其解释为"$\in$"，则通常称之为**标准解释**。

(2)即使在标准解释之下，ZF 与前一公理也不相同。这主要表现在四条公理上，即 $A\varnothing$，AP，AS 和 AR。首先看 $A\varnothing$：前一公理肯定的空集合$\varnothing$一定不含有任何元素；而 ZF 的这条公理肯定的集合，仅不含任何"集合"，至于不是"集合"的实体，这条公理肯定的集合并未说含与否。其次看 AP：前一公理肯定的$P(S)$，地地道道是 S 的所有子集作为元素构成的集合；而 ZF 的这条公理却只能肯定 S 的至多可数个子集，作为所存在的集合的元素。最后看 AS 和 AR：这二条中，一个有"性质 $p(x)$"(或公式 $P(a)$)，一个有"函数 f"(或一阶公式 $A(a,b)$)；对前一公理言，这"$p(x)$，f"是说得含混的，但却蕴含有非可数多个；而对 ZF 言，这"一阶公式 $P(a)$，一阶公式 $A(a,b)$"是说得非常清楚的，但却均只能肯定有可数多个；这差别就使得 AP 在二个不同的公理系统中有了不同的含义了。

以上比较的是二个公理系统的主要不同之处。事实上，这二者也有实质上的相同之处。(1)首先，公理数目相同(忽略性质、函数非可数和可数之分)，而且完全相对应。(2)其次，二者均未限死“集合”，即并未给出“封闭”的话，比如说，“集合仅由上述得到”；所以这样作，我以为是，人们目前并不清楚集合仅由这些就行了，就足够了；事实上，从 AC 和 CH 独立于 ZF 的证明结果，正说明可能存在着人们尚未识识清楚的集合。(3)最后，这二者公理相对应，使得在前一公理中能进行的证明，在 ZF 中也能进行完全相平行的证明，只是在前一公理中论证的对象是非可数多个时，而在 ZF 中论证的对象却可以是可数多个。关于这一点，在第三节中还要加以证明。这便是至今介绍公理集合论的书中，将这二者不加区别地展开讨论的理由。

2-2　序数

从第一节朴素集合论，可见序数是非常重要的。从同构意义上讲，序数的概念讨论清楚了，别的有关集合的概念也就清楚了。正如张锦文在其二部著作中所述，序数是集合论的“**脊梁骨**”、“**精髓**”。分下述三小段讨论。

(1)集合的后继和自然数

设给定集合 x，则称 $x^+=x\cup\{x\}$ 为 x **的后继**(由前述公理，x^+ 当然也是集合)。

若令**0**记$\varnothing$，则可如下定义集合表示的自然数：**0是自然数；如果 x 是自然数，则 x^+ 也是自然数；自然数仅能这样得到。**

由此，0即$\varnothing$，并定义

$$1即0\cup\{0\}=\varnothing\cup\{\varnothing\}=\{\varnothing\}$$

$$2即1\cup\{1\}=\{\varnothing\}\cup\{\{\varnothing\}\}=\{\varnothing,\{\varnothing\}\}$$

$$3即2\cup\{2\}=\{\varnothing,\{\varnothing\}\}\cup\{\{\varnothing,\{\varnothing\}\}\}=\{\varnothing,\{\varnothing\},\{\varnothing,\{\varnothing\}\}\}$$

一般地，任 n 即$(n-1)\cup\{n-1\}=$

$$\{\underset{0个}{\varnothing},\underset{1个}{\{\varnothing\}},\underbrace{\{\varnothing,\{\varnothing\}\}}_{2个},\cdots,\underbrace{\{\varnothing,\{\varnothing\},\cdots,\{\varnothing,\{\varnothing\},\cdots\}\}}_{n-1个}\}。$$

用这些集合来定义自然数是合适的。因为0表示没有任何“东西”，而$\varnothing$也正是没有任何“元素”；1表示只有“一样东西”，而$\{\varnothing\}$正是有一个$\varnothing$作为“唯一元素”；2表示有“二样东西”，而$\{\varnothing,\{\varnothing\}\}$正是有$\varnothing,\{\varnothing\}$作为“二个元素”；一般地，$n$ 表示有“n”样东西”，而$\underbrace{\{\varnothing,\{\varnothing\},\cdots,\{\varnothing,\{\varnothing\},\cdots\}\}}_{n个}$正是有$\varnothing,\{\varnothing\},\cdots,\{\varnothing,\{\varnothing\},\cdots\}$等作为“$n$ 个元素”。另外可见：

$$0\in 1\in 2\in 3\in\cdots;0\subset 1\subset 2\subset 3\subset\cdots$$

这两种性质也正和自然数的涵义相一致。

现在，回头看看无穷公理 $A\omega$ 肯定集合 ω 的存在也是合适的，因为 ω 正是为集合$\{0,1,2,\cdots\}$。顺便说一句，如果没有 $A\omega$，单从其他公理无法肯定无穷集的存在。而有了 ω，事情就好办了。这从后面可见。

(2)归纳集合、传递集合、三歧集合

称一集合 S 是**归纳集合**，如果 S 满足：$0\in S$，且对任集合 x，若 $x\in S$，则 $x^+\in S$。

由定义，显然 ω 是一个归纳集合，且 ω 还是**最小的**归纳集合。即对集合 $T\subset\omega$，且 T 是归纳的，则可证 $T=\omega$（由 ω 的定义和 T 的归纳性，证明任 $x\in\omega$，均有 $x\in T$）。由此，要证命题 $A(n)$对任$n\in\omega$均成立，只要证 $T=\{n|n\in\omega$ 且 $A(n)\}$是归纳的即可。实际上，这就是数学归纳法。

非常著名的 Peano 公理，可表述如下：

$\forall x\in\omega(x^+\neq 0)$，

$\forall x\in\omega\forall y\in\omega(x^+=y^+\rightarrow x=y)$，

$A(0)\wedge\forall x\in\omega(A(x)\rightarrow A(x^+))\rightarrow\forall x\in\omega A(x)$。

在集合论公理下，这三条公理是定理。有兴趣的读者可以去证明之。证明上述第二条时，即要从 $x\cup\{x\}=y\cup\{y\}$得到 $x=y$；但从 $x\cup\{x\}=y\cup\{y\}$，使用 Ae，立得或者 $x=y$，或者 $x\in y$ 且 $y\in x$。利用 AF，知这后一种情况不能成立，故证明。

称一集合 S 是**传递的**，如果 S 的每个元素的元素均是 S 的元素，即

$$\forall x\forall y((x\in y\wedge y\in S)\rightarrow x\in S)。$$

说得再明白一点，传递集合的元素的元素，以及元素的元素的元素等等，均需是该集合的元素。

那么，显然有：S 传递当且仅当1）$\bigcup S\subset S$ 当且仅当2）$\forall x(x\in S\rightarrow x\subset S)$当且仅当3）$S\subset P(S)$。事实上，1）和2）均可看成 S 传递的另二种表述方式，而2）和3）的等价性可如下证明：

2）⇒3）：任 $x\in S$，均有 $x\subset S$，故 $x\in P(S)$；

3）⇒2）：任 $x\in S$，均有 $x\in P(S)$，故 $x\subset S$。

由此，可证下述命题：

命题 1 自然数和 ω 均是传递的集合。

证明 设 A(n)表示"n 是传递的"，则要证任自然数 n，均有 A(n)，只要证 $T=\{n\mid n\in\omega\wedge A(n)\}$是归纳的。∵$0\in T$；且若 $n\in T$（那么 A(n)，即 n 传递），而且∵$n^+=n\cup\{n\}$，∴$\bigcup n^+=\bigcup(n\cup\{n\})=n\subset n^+$，故由传递的等价定义1），$n^+$也传递，即 $n^+\in T$。故 T 是归纳的，因而自然数均是传递的集合。

现证 ω 传递。由$0\in 1\in 2\in 3\in\cdots$，∴$\bigcup\omega=\omega$ 当然$\bigcup\omega\subset\omega$，故同样从等价定义1），$\omega$ 也是传递的集合。故证毕。

命题 2 对任集合 x，若 x 的每一元素均传递，则$\bigcup x$ 也传递。

证明 按定义证。即若 $y\in\bigcup x$，$z\in y$，则有 $z\in\bigcup x$。这是因为，由 $y\in\bigcup x$，即有 $t\in x$，而 $y\in t$，从命题的条件，t 是 x 的元素，t 就传递，从而 $z\in y\in t$，即可得 $z\in t$，再从 $z\in t$，$t\in x$，可得 $z\in\bigcup x$。故证毕。

称一个集合 S 具有"∈"**三歧性**，如果对任 $x,y\in S$，$x\in y$，$x=y$，$y\in x$ 三者中恰有一个成立。分别记作 $x<y$，$x=y$，$y<x$。

那么，由$0\in 1\in 2\in 3\in\cdots$和$0\subset 1\subset 2\subset 3\subset\cdots$知，$\omega$ 具有∈三歧性（∵任 $n,m\in\omega$，设按通常意义下，n 小于 m，那么，$n\in n+1$，而 $n+1\subset n+2$，故 $n\in n+2$，依次可得 $n\in m$，即 $n<m$）。这就是说，在上述所给的"<"（即"∈"）下，ω 的任二元素均可比较。

再从**命题 1**，ω 传递，由等价定义2)，有 $n\in\omega,n\subset\omega$，故也就有 $n<\omega$。这样，任自然数 n 也和 ω 建立了可比关系。

(3)序数

称一个集合 S **为有极大元 x 的**，如果对任何 $y\in S$，只要 $y\neq x$，则恒有 $y\in x$（如按前述记号，即 $y<x$）。

称一个集合 S **为无极大元的**，如果 S 没有任何极大元 x 的话。

现在定义**序数**。称 O **是序数**；且若 α 是序数；也称 α^+ **是序数**；又若 S 是无极大元且满足三歧性的序数集合，则称 $\bigcup S$ **也为序数**。且仅仅由上述得到序数。

由序数的定义，**自然数**是序数。注意到 $\omega=\bigcup\omega$，不难知道，ω **也是序数**。再由序数定义，ω^+ 记为 $\omega+1$，**是序数**；$(\omega+1)^+$ 记为 $\omega+2$，**是序数**；$\cdots(\omega+n)^+$ 记为 $\omega+(n+1)$**是序数**。

命题 3 任一序数均是传递集合。(留作练习)

令 $S_0=\{\omega,\omega+1,\omega+2,\cdots,\omega+n,\omega+(n+1),\cdots\}$，那么：

命题 4 $\bigcup S_0$，记为 $\omega+\omega$，是序数。

证明 ∵有函数 $\{<n,\omega+n>|n\in\omega\}$，∴由 AR，$\mathrm{ran}((f\restriction\omega)=S_0$是集合。而很明显，$S_0$是无极大元且满足三歧性的序数集合，故 $\omega+\omega$ 是序数。

命题 5 1)对任意自然数 n，均有 $n\in\omega+\omega$；

2)对任意自然数 n，均有 $\omega+n\in\omega+\omega$；

3)$\omega+\omega$ 是满足上述1)和2)的最小的序数。即：若 α 是任一序数，且满足1)和2)，则 $\omega+\omega\subset\alpha$。由此，$\omega+\omega=\{0,1,2,\cdots,\omega,\omega+1,\cdots,\omega+n,\cdots\}=\omega\bigcup S_0$。

证明 从$\bigcup S_0$的定义，易知：

$$\omega+\omega=\{0,1,2,\cdots,\omega,\omega+1,\cdots,\omega+n,\cdots\},$$

故证毕。

类似上述，将 $\omega+\omega$ 记为 $\omega\cdot 2$，那么有$(\omega\cdot2)+1$，$(\omega\cdot2)+2$，$\cdots$，将$(\omega\cdot2)+\omega$ 记为 $\omega\cdot3$，那么又有$(\omega\cdot3)+1$，$(\omega\cdot3)+2$，……。

令 $S_1=\{\omega\cdot n|n\in\omega\}$，可得$\bigcup S_1$，记为 $\omega\cdot\omega$，其也为序数，且

$$\omega \cdot \omega = \{(\omega \cdot n) + m \mid m, n \in \omega\},$$

可见 $\omega \cdot \omega$ 是大于形式为$(\omega \cdot n)+m$ 的最小的序数。

命题 6 任一序数 α 均具有三歧性(即$\in$-三歧性)。

证明 施归纳于序数的构造。∵0中无元素，∴0具有三歧性；故仅需对如下二种情况进行归纳证明：

1)若 β 具有三歧性，则 β^+ 也具有三歧性。即若 $x,y \in \beta$ 时，$x<y, x=y$ 或 $y<x$ 中恰有一个成立，而 $\beta^+ = \beta \cup \{\beta\}$，故若 $x,y \in \beta^+$，则可能是 $x,y \in \beta$，自然不用证，也可能是 $x \in \beta$，而 $y=\beta$ 或者是 $y \in \beta$，而 $x=\beta$，此时，自然 $x \in y$ 或 $y \in x$，即 $x<y$ 或 $y<x$，还有可能是 $x=y=\beta$，自然更不用证(这里，均使用了：当 $x<y, x=y$ 或 $y<x$ 成立，则必恰有一个成立。利用 AF。是易得到证明的。留作练习)。

2)设 S 是无极大元且满足三歧性的序数集合(自然任 $\alpha \in S$，α 也满足三歧性)，则$\bigcup S$ 也满足三歧性。设 $x,y \in \bigcup S$，现证 $x<y, x=y$ 或 $y<x$ 恰有一个成立。由 $x,y \in \bigcup S$，则必有 $\alpha, \beta \in S$，且 $x \in \alpha, y \in \beta$，由 S 满足三歧性，则 $\alpha \in \beta, \alpha=\beta$ 或 $\beta \in \alpha$ 恰有一个成立。当 $\alpha \in \beta$ 时，则有 $x \in \alpha \in \beta, y \in \beta$，从序数的传递性，也有 $x \in \beta$，故从归纳设(任 $\alpha \in S$，α 满足三歧性)，$x<y, x=y$ 或 $y<x$ 恰有一个成立。当 $\beta \in \alpha$ 时，类似可证结论。而当 $\alpha=\beta$ 时，自然更可证结论。该命题证毕。

命题 7 对于任集合 x，若它是传递的且又具有三歧性，则 x 是一个序数。

证明 用反证法。即有一集合 x，其传递且具三歧性，但 x 非序数。那么，令

$S=\{y \mid y$ 传递、且具三歧性、又非序数，$y=x$ 或 $y \in x\}$那么，S 非空(∵$x \in S$)，故由 AF，即有一 $x_0 \in S$，使 $x_0 \cap S = \varnothing$。且由 x_0非序数，故 $x_0 \neq \varnothing$，即有 $y \in x_0$。然后，从 x_0的性质又导致 x_0必为一序数。矛盾，故证。

由命题 7，即得序数的另一等价定义。这即 von Neumann 于1925年所给的定义。

称一序数 α **为后继序数**，如果有序数 β，使 $\alpha=\beta^{+}$。而非0，非后继序数的序数，称为**极限序数**。

下述**命题 8**中，仅列出序数的其他一些性质，不再给出证明。有兴趣的读者，可按所列次序逐一证明之。读者也可参考张锦文的二部著作。

命题 8 1〉ω 是极限序数；

2〉对任序数 α，不存在序数 β，使 $\alpha<\beta<\alpha^{+}$；

3〉任序数 α 的 **β 前节 $\mathrm{Seg}(\beta)$**$=\{\gamma|\gamma<\beta,\gamma\in\alpha\}$是序数；

4〉任序数 α,β，如果 $\alpha\in\beta$，则 $\alpha\subset\beta$；

5〉任序数集 S 的极大元唯一。由此，序数集 S 的极大元也可称为其**最大元**；

6〉任序数集 S，$d\in S$ 且为 S 的最大元，则$\bigcup S=d$；

7〉任序数集 S，如 S 无最大元，且 $S\neq\varnothing$，则$\bigcup S$ 为极限序数；

8〉任序数 α,β，如果 $\alpha\subsetneq\beta$，则 $\alpha\in\beta$；

9〉任序数集 S 具有三歧性；

10〉任集合 S，那么：x 是序数当且仅当 x 传递，且任 $y\in x$，y 也传递。

由上述**命题**8的9〉，任何两个序数均可比较。而由10〉，我们又得到序数的一个等价定义。这样，已介绍了序数的三个等价定义。

2.3 序数和基数

本小节的目的，是要得到所有的序数。并同时给出基数的定义，也与序数相辅地得到。

称序数 α 是**可数无穷序数**（或简作**可数序数**），如果有 ω 到 α 的双射函数。

易知，$\omega,\omega+n,\omega+\omega$ 均为可数序数。后者的可数性由双射 f 可见。f 定义为：

$$f(n)=\begin{cases}k,n=2k\\ \omega+k,n=2k+1,\end{cases}$$

即 $f(0)=0,f(1)=\omega,f(2)=1,f(3)=\omega+1,f(4)=2,\cdots$。

对任意的序数 α,β，如果有1-1函数 $f:\alpha\to\beta$，则记作 $\overline{\overline{\alpha}}\leqslant\overline{\overline{\beta}}$；而如果没有双射函数 $f:\alpha\to\beta$，则称 α **与** β **不等势**，记作 $\overline{\overline{\alpha}}\neq\overline{\overline{\beta}}$；当 $\overline{\overline{\alpha}}\leqslant\overline{\overline{\beta}}$，且又有 $\overline{\overline{\alpha}}\neq\overline{\overline{\beta}}$ 时，称 α **的势小于** β **的势**，记作 $\overline{\overline{\alpha}}<\overline{\overline{\beta}}$。

序数 α 称作**基数**，如果对任何序数 $\beta<\alpha$，均有 $\overline{\overline{\beta}}<\overline{\overline{\alpha}}$。并定义**基数的次序**为相应的序数之次序。

那么易知：自然数 n，序数 ω 均是基数。

定理 令 $\omega_1=\{\alpha|\alpha$ 为序数，且 $\overline{\overline{\alpha}}\leqslant\overline{\overline{\omega}}\}$，则 ω_1**是基数**。

证明 该定理证明很长，分下述几步。

1〉首先，定义**良序关系**。

设 R 是集合 S 上的二元关系。称 R **满足传递性**，如果对任 x，$y,z\in S$，由 xRy 和 yRz 可推得 xRz；而称 R **满足三歧性**，如果对任 $x,y\in S$，下述三者恰有一个成立，即 xRy，$x=y$ 或 yRX。称 $u\subset S$ 中的 x **为** u **中关系** R **的首元素**，如果对任 $y\in u$，yRx 均不成立。

称集合 S 上的二元关系 R 是**良序关系**，如果 R 满足传递性、三歧性（这二者即全序关系），且 S 的任不空子集 u，均有 x 为 u 中关系 R 的首元素。

2〉**良序集**，**良序结构**和序数。

称集合 S 是**良序集**，如果有 S 上的二元关系 R 是良序关系。此时，也称该 R **将** S **给良序了**。并称 $\langle S,R\rangle$ 为一**良序结构**。

那么，不难知道，任序数 α，均是由"$\in$"良序的良序集。反之，任一传递的集合 S，当其又是由"$\in$"良序的良序集时，则 S 就是一序数。

当定义了良序结构的**同构**之后（这里略去），不难证明：任一良序结构 $\langle S,R\rangle$，都有唯一的序数 α，使 $\langle S,R\rangle$ 与 $\langle\alpha,\in\rangle$ 同构。

另外，对良序集 S，设 $T\subset S$，定义 $\sup T$ 为 $(S-T)$ 的首元素；特别地，$\sup\varnothing$ 为 S 的首元素。

3〉设 R 是 ω 上的良序关系，则 $R\subset\omega\times\omega$，故由 AS，R **为一集合**；

令 $M=\{R|R$ 是 ω 上的良序关系$\}$，

则易知：$M \subset P(\omega\times\omega)$，即有 $M\in P(P(\omega\times\omega))$，因此，再使用 AS，M 也为一集合。

4）ω_1**是集合**。这只要作一函数 f，使 $\omega_1=\mathrm{ran}(f\upharpoonright M)$即可（这是∵$M$ 是集合，∴使用 AR 可得 ω_1是集合）。为此，又分下述三步：

(*a*)作 f_R，其定义域为 $\mathrm{ran}R=\{y\mid y\in\omega$ 且有 $x\in\omega$，使 $xRy\}$，且对任 $x\in\mathrm{ran}R$，令 $f_R(x)=\sup\{y\mid y\in\omega$ 且 $yRx\}$。现在举例看看函数 f_R，

设$\langle\omega,R\rangle$为$0,2,4,6,8,\cdots,1,3,5,7,9,\cdots$

那么，$\mathrm{ran}R=\omega-\{\phi\}$，记为 ω^*，且

$f_R(2)=\sup\{y\mid y\in\omega$ 且 $yR2\}=\sup\{0\}=2$，

$f_R(4)=\sup\{y\mid y\in\omega$ 且 $yR4\}=\sup\{0,2\}=4$，

⋮

$f_R(1)=\sup\{0,2,4,6,\cdots\}=1$

$f_R(3)=\sup\{0,2,4,6,\cdots,1\}=3$

⋮

故 $\mathrm{ran}f_R=\{2,4,6,\cdots,1,3,5,\cdots\}$。

而且，一般而言，$f_R(x)$总是一序数，且 $\mathrm{ran}f_R$ 也是由 R 唯一确定的序数。

(*b*)作函数 f，其定义域为 M，且对任给的 $R\in M$，

$$f(R)=\mathrm{ran}f_R,$$

易知，$f(R)=\in w_1$。

(*c*)现在证：对任给 $\alpha\in\omega_1$，均有 $R\in M$，使 $f(R)=\alpha$。

∵$\alpha\in\omega_1$，即 $\alpha\leqslant\omega$，亦即 α 有穷或可数。现仅证 α 是可数的情况，即$\overline{\overline{\alpha}}=\overline{\overline{\omega}}$；此时，有一双射 $g:\omega\to\alpha$，即对任 $\beta\in\alpha$，有 $n\in\omega$，使 $g(n)=\beta$；现在定义 ω 上的关系 R：对任 $n_1,n_2\in\omega$，n_1Rn_2当且仅当 $g(n_1)\in g(n_2)$；由 α 的传递性、三歧性可得 R 的传递性和三歧性，且$\langle\omega,R\rangle$的任一不空子集也有极小元（即任 $u\subset\omega$ 中，有 x 为 u 中关系 R 的首元素）；故 $R\in M$，而 $f(R)=\alpha$。

由上述，使用 AR，可得 ω_1是集合。

5）ω_1**是基数**。为此，先证 ω_1是序数，而这只要证 $\omega_1=\bigcup\omega_1$即

可。事实上，ω_1是传递的：对任 $y\in\omega_1$，$x\in y$，∵ y，x 均为序数，且 $x\subset y$，故有 $\overline{\overline{x}}\leqslant\overline{\overline{y}}\leqslant\overline{\overline{\omega}}$，∴ $x\in\omega_1$；ω_1也无最大元：若不然，设 $\alpha\in\omega_1$，α 为 ω_1的最大元，且 $\overline{\overline{\alpha}}\leqslant\overline{\overline{\omega}}$，但$\overline{\overline{\alpha+1}}\leqslant\overline{\overline{\omega}}$，且 $\alpha<\alpha+1$，与 α 为最大元矛盾；这样，由**命题 8** 中序数的性质7），$\bigcup\omega_1$是一极限序数；结合 ω_1的传递性，$\bigcup\omega_1=\omega_1$。故 ω_1是基数。定理证毕。

且显然有，ω_1不可数。∴ ω_1是最小的不可数基数。

类似地，可构造更大的不可数基数。这样，可把无穷基数排成：

$$\omega_0,\omega_1,\omega_2,\cdots,\omega_\omega,\cdots,\omega_\alpha,\cdots$$

而文献中常写作：

$$\aleph_0,\aleph_1,\aleph_2,\cdots,\aleph_\omega,\cdots,\aleph_\alpha,\cdots。$$

2.4 超穷归纳法

设 $\lim(\alpha)$表示：α 是极限序数，并设 $T(\mu)$是一个有关序数的命题，那么，可用下述二种形式表示超穷归纳法。

$$(1)T(0)\wedge\forall\alpha(T(\alpha)\rightarrow T(\alpha^+))\wedge\forall\alpha(\lim(\alpha)\wedge\forall\beta\in\alpha T(\beta)\rightarrow T(\alpha))\rightarrow\forall\alpha T(\alpha);$$

$$(2)\forall\alpha(\forall\beta\in\alpha T(\beta)\rightarrow T(\alpha))\rightarrow\forall\alpha T(\alpha)。$$

上述两种形式是用逻辑语言表述的，读者很明白，不多作解释了。

第三节　ZF 系统

公理集合论研究始于1904年的 Zermelo。当时，他提出选择公理 AC，并用它证明了良序定理。这是按 Hilbert 的建议企图证明 CH 的。1908年，Zermelo，Russell 分别提出两个看起来很不相同的集合论公理系统。但到后来，经不少人发展，特别是经 Skolem 等人，最终于1928年，Fraenkel 构成了现在的 ZF 系统。本节，就是要介绍 ZF 系统。在第二节，已对比前一公理系统，引入了 ZF 的各条公理，并评论了二者的异同。这里，仅对公理再作一说明，并不需展开有关序数等的讨论。然后，侧重介绍选择公理的几种等价形

式，并给出部分证明，以飨读者。最后，粗略地从思想上介绍 AC、CH 独立于 ZF 的证明。

3.1 公理和说明

已如**2.1**中所给出的九条公理：$A\phi$，Aw，AB，AU，AP，AS，AR，Ae 和 AF。其中，AS 中的 $P(a)$和 AR 中的 $A(a,b)$均为一阶公式，或用第三章的记号，即 $P(a)$，$A(a,b)\in L_I^S$，而 $S=\{\in\}$。这样，在第二节先给出的公理系统中，没有说清楚的 $p(x)$和函数 f，现在均说得非常严格，或非常"形式"了。但正如**2.1**中所说，经这种限定后，对一些本该是集合的，却划在了可以是，也可以不是的状态了。特别是幂集公理 AP，其肯定存在的集合，既可以是朴素集合论中的幂集，也可以是可数个子集构成的集合(不是真正意义上的幂集了)。

当然，按第二节在公理集合论中讨论序数的过程，在 ZF 中可同样展开讨论，讨论中所用的语言相似(即将那里的自然语言，换成一阶语言)，只不过所讨论的"内容"(或称"对象")却不相同。若用第三章的术语言之，即在 ZF 中，讨论的对象包含更多的非标准模型。显然，无需再进行这种讨论。用句通俗的话说：二个公理系统正像演戏，戏台一样，而上演的戏却不同。

3.2 选择公理 AC

这里仅列出 AC 的九种等价形式，并在 ZF 中给出其中七种相互等价的证明。

AC_1(单值化原则)：对任关系 R，均有一函数 h，使 $\mathrm{dom}(R)=\mathrm{dom}(h)$，且 $h\subset R$。也可用符号严格写为：$\forall x(\mathrm{Re}(x)\rightarrow\exists y(\mathrm{Fun}(y)\wedge\mathrm{dom}y=\mathrm{dom}x\wedge y\subset x))$。

AC_1(乘积定理)：对任集合 I 和任意一个定义域为 I 的函数 g，如果对 $i\in I$，都有 $g(i)\neq\varnothing$，则 $\prod_{i\in I}g(i)\neq\varnothing$。其中，

$$\prod_{i\in I}g(i)\underset{df}{=}\{f\mid\mathrm{Fun}(f)\wedge\mathrm{dom}(f)=I\wedge\forall i(i\in I\rightarrow f(i)\in g(i))\}。$$

Cohen 的书上就是这种形式。也可用符号严格写为：$\forall I\forall g$ $(\mathrm{Fun}(g)$ $\wedge \mathrm{dom}g=I\wedge\forall i\in I\forall x(\langle i,x\rangle\in g\rightarrow x\neq\varnothing)\rightarrow\exists f(\mathrm{fun}(f)\wedge \mathrm{dom}f=$ $I\wedge\forall i\in I\forall y\forall z(\langle i,y\rangle\in f\wedge\langle i,z\rangle\in g\rightarrow y\in z)))$。

AC_{III}（序数形式的 AC）：对任集合 S，都有一个序数 α 和 α 上的函数 g，使对该 S 满足：$y\in S$ 当且仅当有 $\beta\in\alpha$，且 $y=g(\beta)$（即 $S=\{g(\beta)\mid\beta\in\alpha\}$）。也可用符号严格写为：$\forall x\exists\alpha\exists g(On(\alpha)\wedge$ $\mathrm{Fun}(g)\wedge\forall y\in x\exists\beta\in\alpha(\langle\beta,y\rangle\in g))$

AC_{IV}（采样原则）：对任一由不空的且两两不相交的集合组成的集合 x，我们总可以从其不空的两两不相交的集合中同时恰好各取一个元素，组成一新集合 C。也可用符号严格写为：$\forall x[\forall y\in x$ $(y\neq\phi\wedge\forall y_1,y_2\in x(y_1\neq y_2\rightarrow y_1\cap y_2=\varnothing)\rightarrow\exists c(\forall z(z\in x\rightarrow\exists! t(t\in c$ $\wedge t\in z)\wedge\forall t(t\in c\rightarrow\exists! z(z\in x\wedge t\in z)))]$

AC_{V}（势的三歧性原则，该形式由 Cantor 提出）$\langle Ca,<\rangle$ 是全序的。其中，Ca 是全部基数，即任二集合 S_1,S_2，下述三者恰有一个成立：$\overline{\overline{S_1}}<\overline{\overline{S_2}}$，$\overline{\overline{S_1}}=\overline{\overline{S_2}}$，$\overline{\overline{S_2}}<\overline{\overline{S_1}}$。

AC_{VI}（良序定理）：任一集合均可良序。该形式也可用符号严格写为：

$$\forall S\exists x(x\subset S\times S\wedge\forall y(y\subset S\wedge y\neq\phi\rightarrow\exists z(z\in y\wedge\forall t\in y(\langle t,z\rangle\notin x)))).$$

AC_{VII}（Zorn 引理）：偏序集 $\langle S,R\rangle$ 的子集 C 称为 S 的一个链，如果 $\langle C,R\rangle$ 是线序；称 u 是链 C 的一个上界，如果对每一 $x\in C$，都有 xRu；称 $t\in S$ 是 S 的极大元，如果不存在 $x\in S$，使 tRx。那么，Zorn 引理论断：若 S 的每链上都有一上界，则 S 有极大元。

AC_{VIII}（Zermelo 的 AC）：对任意的非空集 S，都存在一个函数（称之为对 $P(S)-\{\varnothing\}$ 的选择函数），使得对于任意非空的 $x\subset S$，都有：$f(x)\in x$（或陈述为：由非空集组成的集族 S，都有选择函数 f，使对任 $x\in S$，均有 $f(x)\in x$）。

AC_{IX}（Tukey 引理）：设 F 为诸集合作成的非空类，如 F 具有有限特征，则 F 含极大元。

在朴素集合论讨论中，曾用 AC_{VIII} 证过 AC_{VI}，即用选择公理证

良序定理。现在，将作下述论证：

$$AC_{I} \overset{(1)}{\Rightarrow} AC_{II} \overset{(2)}{\Rightarrow} AC_{IV} \overset{(3)}{\Rightarrow} AC_{VIII} \overset{(4)}{\Rightarrow} AC_{III} \overset{(5)}{\Rightarrow}$$

$$AC_{VII} \overset{(6)}{\Rightarrow} AC_{VI} \overset{(7)}{\Rightarrow} AC_{I}$$

(1) $AC_{I} \Rightarrow AC_{II}$：这即假定 AC_{I} 成立。为证 AC_{II} 成立，即设有 g, I 满足 AC_{II} 的前提，$\mathrm{dom}(g)=I$，如果对任 $i\in I, g(i)\neq\varnothing$，而后证 AC_{II} 的结论：即有 $f\in\prod_{i\in I}g(i)\neq\phi$。由 I 和 g，定义关系

$$R=\{\langle i,x\rangle \mid i\in I\wedge x\in g(i)\}$$

（要使用 AC_{I}，必须定义 R），故由 AC_{I}，有函数 $f\subset R$，且 $\mathrm{dom}(f)=\mathrm{dom}(R)=I$。现证：该 f 即为 $\prod_{i\in I}g(i)$ 中的。$\because \langle i,f(i)\rangle\in f\subset R=\{\langle i,x\rangle \mid i\in I\wedge x\in g(i)\}$

$\therefore f(i)\in g(i)$。因而证毕。

(2) $AC_{II}\Rightarrow AC_{IV}$：设集合 X 满足 AC_{IV} 的前提，即 X 的每个元都不空，且 X 的任二不同元均不相交，现证 AC_{IV} 的结论成立。令 H 是 x 上的一个恒等函数，即对任 $y\in x$，均有 $H(y)=y$。因此，有：$\forall i\in x(H(i)\neq\phi)$。这即构成 AC_{II} 的条件；因而，有 $f\in\prod_{i\in x}H(i)=\{f \mid \mathrm{Fun}(f)\wedge\mathrm{dom}(f)=x\wedge\forall i(i\in x\to f(i)\in H(i))\}$，令 $C=\mathrm{ran}(f)$，那么，对任一 $i\in x$，$\{i\}\cap C=\{f(i)\}$。故这 C 即 AC_{IV} 中所求。

(3) $AC_{IV}\Rightarrow AC_{VIII}$：设非空集 S（这即 AC_{VIII} 的前提），现在构造 $P(S)-\{\phi\}$ 的选择函数 f，满足 AC_{VIII} 的结论。由 S，造 $T=\{\{b\}\times b \mid b\subset S$，且 b 非空$\}$，那么，T 的每个元素均非空，且 T 的任二不同元均不相交，所以，T 满足 AC_{IV} 的前提，故可令 c 是由 AC_{IV} 得到的那个集合，该 c 与 T 的每个元素的交集自然是一个单元集合（即只含一个元素的集合），即 $c\cap(\{b\}\times b)=\{\langle b,x\rangle\}$，而 $x\in b$。现在，令 $f=c\cap(\bigcup T)$，我们来证：该 f 即所求。因为，f 的任一元素都是属于某一 $\{b\}\times b$（$b\subset S$，非空）的，而对任 $x\in b$，f 的元素形如 $\langle b,x\rangle$，且对任非空的 $b\subset S$，均有唯一的 x，使 $\langle b,x\rangle\in f$。故证毕。

(4) $AC_{VIII}\Rightarrow AC_{III}$：任给一集合 S，由 AC_{VIII}，有一选择函数 g，使

对任不空集合 $x \subset S$，均有 $g(x) \in x$。从该 g 出发，令（按超穷归纳定义）

$$S_0 = S,$$
$$S_{\beta+1} = S_\beta - g(S_\beta),$$
$$S_\lambda = S - \bigcup_{\beta<\lambda} g(S_\beta)\text{，当 }\lambda\text{ 为极限序数时。}$$

可见，对任意不同的序数 β_1, β_2，均有 $S_{\beta_1} \neq S_{\beta_2}$，并且 $S_\beta - S_{\beta+1}$ 恰有一个元素，所以总有一序数 α，使 $S_\alpha = \phi$。我们取满足这一性质的最小序数 α，则这 α 和 f（该 f 定义为 $f(\beta) = g(S_\beta)$，$\beta \in \alpha$）就是 AC_{V} 中要求的 α 和 f。证毕。

(5) $AC_{\mathrm{V}} \Rightarrow AC_{\mathrm{VI}}$：对任给的集合 S，由 AC_{V}，即有序数 α 和 α 上的函数 g，且 $S = \{g(\beta) \mid \beta \in \alpha\}$。要证 AC_{VI}，即证 S 均可良序。利用所得的 α 和 g，令 $a_\beta = g(\beta)$，$\beta < \alpha$
则 $S = \{a_0, a_1, \cdots, a_\beta, \cdots\}$，$\beta < \alpha$，且这 S 即与 α 建立了一个同构对应，从而在 S 上建立了关系"$<$"；这样，对 S 的任子集 S_1，也都恰对应于 α 的相应的子集 α_1，因 $\alpha_1 (\subset \alpha)$ 有最小元 γ，故可取 a_γ 为 S_1 的最小元。结果，S 得到了良序。证毕。

(6) $AC_{\mathrm{VI}} \Rightarrow AC_{\mathrm{VII}}$：设 $\langle S, R \rangle$ 是不空的偏序集，且 S 的每链都有上界，现在证 S 有极大元。对所给的 S，使用 AC_{VI}，可得一良序结构 $\langle S, < \rangle$；现在，按超穷归纳定义 $f: S \to \{0, 1\}$，使对任何

$$a \in S, f(a) = \begin{cases} 1\text{，当 } \forall b (b \in S \wedge b < a \wedge f(b) = 1 \to bRa) \\ 0\text{，否则，} \end{cases}$$

（$\because \langle S, < \rangle$ 是良序结构了，故可从最小的 $a \in S$，开始定义，当然 $f(0) = 1$，而对任 $a > 0$，均可按上述全部定义出；即对任 $a \in S$，$f(a)$ 均有了定义）
由此 f，定义集合 $C = \{a \mid a \in S \wedge f(a) = 1\}$，那么，$C$ 当然是在 R 下的链，故 C 有上界 t。现在证明，这 t 即是 S 的极大元：若不然，即有 $a \in S$，且 $a \neq t$，且 tRa；但另方面，对任 $b \in C$，均有 bRt（$\because t$ 是 C 的上界），故从 bRt，tRa 立得 bRa；因此，由 f 的定义，就有 $f(a) = 1$，$\therefore a \in C$，故又得 aRt，这与 tRa 相矛盾。故证毕。

(7) $AC_{\mathrm{VII}} \Rightarrow AC_{\mathrm{I}}$：即使用 Zorn 引理，证明对任关系 R，均有函

数 F，$F\subset R$ 且 $\mathrm{dom}(F)=\mathrm{dom}(R)$。如下：

令 $\quad x=\{f|f\subset R\ \&\ \mathrm{Fun}\ (f)\}$，则 x 为偏序结构 $\langle x,\subset\rangle$，且易证 x 对其链之"$\bigcup$"封闭(即设 $C\subset x$，C 是对"$\subset$"关系的一个链，则 $\bigcup C\in x$；若 $\langle x,y\rangle\in\bigcup C$，$\langle x,z\rangle\in\bigcup C$，则由"$\bigcup$"的定义，有 f_1，$f_2\in C$，使 $\langle x,y\rangle\in f_1$，$\langle x,z\rangle\in f_2$，再由 C 是对"$\subset$"关系的链，故 $f_1\subset f_2$ 或者 $f_2\subset f_1$，因而无论如何，均有 $y=z$，故 $\bigcup C$ 也是一个函数；另方面，对任 $t\in\bigcup C$，必有 $f\in C$，$t\in f$，因而，结合 $C\subset x$，可知 $t\in R$，$\therefore\bigcup C\subset R$。又由 x 的定义，$\bigcup C\in x$。故 x 对其链之"$\bigcup$"封闭证毕)。也易证 x 的任链 C 的上界即为 $\bigcup C$。这样，即可使用 Zorn 引理，x 有极大元 F，且 $F\subset R$($\because F\in x$)，$\mathrm{dom}(F)=\mathrm{dom}(R)$($\because$ 若 $\mathrm{dom}(F)\neq\mathrm{dom}(R)$，则有 $z\in\mathrm{dom}(R)-\mathrm{dom}(F)$，那么就有 $y\in\mathrm{ran}(R)$，且 zRy，由之定义 $F'=F\bigcup\{\langle x,y\rangle\}$，且由 x 之定义，$F'\in x$，这与 F 的极大性相矛盾)。因而，证毕。

3.3 集合全域(或称集合的论域)

所谓**集合全域**，即集合公理的模型之论域。对朴素集合论，当然无所谓集合全域了。而第二节介绍的前一个集合论公理系统，其全域也是没有限死的，即也可以有各种各样的集合全域，只不过这种全域中一定会有真正的不可数集合。当然，若修改那里的公理，比如说，增加一条公理，其含义是"集合仅仅由那些存在性公理所得到"(即所谓的"封闭性")，则那里的集合全域也就限死了。诚如在**2.1**末的评论中所说，"*AC* 和 *CH* 独立于 *ZF* 的证明结果，正说明可能存在着人们尚未认识清楚的集合"。这就是不加"封闭性"的原因。

ZF 的集合全域当然更是活的了，而且这种全域中甚至可以没有真正的不可数集合。本节所讨论的集合全域，也就是单指 *ZF* 的集合全域。

经常用 V 表示 *ZF* 的集合全域。那么，V 上的结构只有"$=$"和"$\in$"。这里的"$=$"即通常涵义下的"相等"，而"$\in$"不过是由 *ZF* 公理所限定的二元关系，其相应于二元关系符号"$\in$"。

设给定了某个集合全域 V'（即一个固定的 V）。我们以三条公理为例，解释一下什么叫"V'满足该公理"，或者"该公理在 V' 中为真（成立）"。空集公理 $A\phi$ 为：$\exists x \forall y(\neg y \in x)$。所谓 V' 满足 $A\phi$，或 $A\phi$ 在 V' 中为真，即是说"在 V' 中有一 x，使 V' 中的任何 y，y 和 x 都无关系"$\in$""可用记号写作：

$$\exists x \circledin V' \forall y \circledin V' (\neg y \in x),$$

亦即 $\exists x(x \circledin V' \wedge \forall y(y \circledin V' \Rightarrow \neg(y \in x)))$；

而公理 AU 在 V' 中成立，即

$$\forall x \circledin V' \exists y \circledin V' \forall u \circledin V' (u \in y \Leftrightarrow \exists z \circledin V' (z \in x \wedge u \in z)),$$

亦即 $\forall x(x \circledin V' \Rightarrow \exists y(y \circledin V' \wedge \forall u(u \circledin V' \Rightarrow (u \in y \Leftrightarrow \exists z(z \circledin V' \wedge (z \in x \wedge u \in z))))))$；

再如公理 AP，它在 V' 中成立，即

$$\forall x \circledin V' \exists y \circledin V' \forall u \circledin V' (u \in y \Leftrightarrow \forall t \circledin V' (t \in u \Rightarrow t \in x)),$$

亦即 $\forall x(x \circledin V' \Rightarrow \exists y(y \circledin V' \wedge \forall u(u \circledin V' \Rightarrow (u \in y \Leftrightarrow \forall t\ (t \circledin V' \Rightarrow (t \in u \Rightarrow t \in x)))))))$。

当把"$\in$"就解释为 V' 中的"属于"（即标准解释）时那么 AP 即：$x \circledin V' \Rightarrow y = \{u \mid u \circledin V' \wedge u \subseteq x\} \circledin V'$；可见，当 V' 中若仅有可数个个体时，尽管 x 中有无穷个个体（此时 x 的子集有不可数多个），但所得的 y 中也仅有可数个个体。

总之，正如 L-S-T 定理（第三章）所述，一阶逻辑不能刻划住无穷模型（或不能抓住无穷模型），故上述 ZF 的全域 V 也就是各种各样的了，只要其能满足 ZF 的各条公理即可。

这里再作一点注解。ZF 的公理 AS 中，有"$P(a)$ 是一阶公式"；在 AR 中，也有"$A(a,b)$ 是一阶公式"之说。这就是说，$P(a)$，$A(a,b) \in L_I^S$，，而 $S = \{\in\}$。不难知道，一阶公式还是可以扩充的。当然，扩充之后，并未改变原来的语义涵义。比如说，$B(a,b,c) \in L_I^S$；当由 ZF 可肯定其有个体 $\boldsymbol{a} \circledin V$ 时（即无论 V 怎样不同，总会有这个体 $\boldsymbol{a}$），即可将 $B(a,b,\boldsymbol{a})$ 也扩充为一阶公式；实际上，即 $B(a,b,\boldsymbol{a}) \in L_I^{S'}$；而 $S' = \{\in, \boldsymbol{a}\}$。可见扩充一阶公式，不会改变原来的语义涵义。因而，在 ZF 的公理 AS，AR 中，允许 $P(a)$，$A(a,b)$

为扩充的一阶公式。

3.4 ZF 的可构成模型 L，ZF 与 AC、GCH 的协调性

这一小节，只能作个概括介绍。但也能使读者了解证明过程和方法。一些细节省略了。

(1)广义连续统猜想 GCH 即：$\overline{\overline{P(\omega_\alpha)}}=\omega_{\alpha+1}$，$\alpha$ 为序数。当 $\alpha=0$时，即 CH。在朴素集合论中曾表述过 CH，而公理集合论(包括 ZF)也可有类似的表述。不过，由于公理集合论(包括 ZF)中，集合全域是各种各样的，故以上的表述并不很确切。而应换成以命题(抽象形式)表示，比如，CH 可表示为：

$$\exists f[\underset{1-1}{\text{fun}}(f)\wedge\exists y(\neg\text{con}(y)\wedge O_n(y)\wedge\neg\text{num}\,b(y)$$
$$\wedge\forall z(\neg\text{con}(z)\wedge O_n(z)\wedge\neg\text{numb}(z)$$
$$\to y\leqslant z))\wedge\forall x(x\in P(w)\leftrightarrow\exists z(f(x)$$
$$=z\wedge z\in y)],$$

其中，$\underset{1-1}{\text{fun}}(f)$表示"$f$ 是$1-1$的函数"，$\text{con}(y)$表示"y 可数"，$\text{numb}(y)$表示"y 是自然数"。这些是可以全用 $L_I^S(S=\{\in\})$中的公式表示出来。有兴趣的读者试补出来，并请试练习写出 GCH。

当用这种抽象的命题表示了 GCH，且尽管在各种集合论中均"形式上"相同，但其涵义却是有所不同的。在朴素集合论中，这种表示就是 Cantor 的涵义，而在公理集合论(包括 ZF)中，因其中变量所指的，是全域并不确定的个体，故彼此有实质上的差别。特别是在 ZF 中，其连"可数"和"非可数"都无法区别，又如何能确切表达 CH 和 GCH 呢？但是，正是由于这种"形式"上相同，使得在 ZF 证得的独立性结果，在其他公理集合论中也同样能成立。换句话说，要想不加新的公理，试图去证已经证明的独立性结果，那是不可能办到的。这就是 ZF 的重要贡献。

(2)定义可构成模型 L

可构成模型 L，是 Gödel 证明 ZF 与 AC、GCH 协调时而构造的 ZF 的模型。这种模型是内模型，正如 Henkin 证明逻辑完全性

所用的方法。当然，Gödel 是在1939年作这件事的，而 Henkin 则是在1949年。这二人的开创性工作构成了现代模型论的一个重要研究方向。

设 $A^0, A^1, A^2, \cdots$ 是扩充的一阶公式的枚举。

若 X 为集合，则令 $X' = X \cup X^+$，

而 $X^+ = \{y \mid \exists n \in \omega, \exists t_1, \cdots, t_{k_n} \in X$，使 $y = \{z \in X \mid A_X^n(z, t_1 \cdots, t_{k_n})\}\}$，其中，$A_X^n(a, b_1, \cdots, b_{k_n})$ 是相对于 X 的公式 $A^n(a, b_1, \cdots, b_{k_n})$。这里解释一下"相对于 X 的公式 $A^n(a, b_1, \cdots, b_{k_n})$"。

对任给 ZF 的个体 X，将 $\forall vB, \exists vB$ 中的量词分别作如下限定，即将 $\forall vB, \exists vB$ 分别改为

$$(\forall vB)_X \underset{df}{=} \forall v(v \in X \to B_X),$$
$$(\exists vB)_X \underset{df}{=} \exists v(v \in X \wedge B_X),$$

其中，B_X 也假定是已作过相应改变的了。那么，称 $(B)_X$ 为**"相对于 X 的公式 B"**。

那么，从 X 得到的 X'，满足下述定理。

引理3.1 1) $X' \subseteq P(X) \cup X$；

2) X' 为集合；

3) X 若为无穷集，且假设 AC，则 $\overline{\overline{X'}} = \overline{\overline{X}}$。

证明　仅需证 X' 为集合。

由 AS，$y_n = \{z \mid z \in X \wedge A_X^n(z, t_1, \cdots, t_{k_n})\}$ 为集合；

由 AR，$\{y_n \mid n \in \omega\}$ 也为集合；

再使用 AU，$\bigcup_{n \in \omega} y_n = \bigcup \{y_n \mid \eta \in \omega\}$ 也是集合，

而 $X^+ = \bigcup_{n \in \omega} y_n$，故 X' 是集合。证毕。

现在，构造可构成模型 L。

令 $$\begin{cases} M_0 = \varnothing \\ M_\alpha = \bigcup_{\beta < \alpha} (M_\beta)', \end{cases} \qquad L = \bigcup_\alpha M_\alpha。$$

为更具体地看到 L 的构成，如下计算一些 M_α：

$$\begin{cases} M_0 = \varnothing \\ M_0^+ = \{y = \{z \in M_0 \mid A\}\} = \{\{z \in \varnothing \mid A\} = \varnothing\} = \{\varnothing\} = P(M_0) \\ M'_0 = M_0 \cup M_0^+ = \varnothing \cup \{\varnothing\} = \{\varnothing\} = P(M_0) \end{cases}$$

$$
\begin{cases}
M_1 = M'_0 = \{\varnothing\} \\
M_1^+ = \{y = \{z \in M_1 \mid z = z\} = M_1, y = \{z \in M_1 \mid z \neq z\} = \varnothing\} \\
\qquad = \{\{\varnothing\}; \varnothing\} = P(M_1) \\
M'_1 = M_1 \cup M_1^+ = \{\varnothing, \{\varnothing\}\} = P(M_1)
\end{cases}
$$

$\vdots$

$$
\begin{cases}
M_\omega = (M_0)' \cup (M_1)' \cup (M_2)' \cup \cdots \\
\qquad = \{\text{从}\varnothing\text{开始构造的所有有穷集}\} \\
M_\omega^+ = \{M_\omega, \varnothing, y = \{z \in M_\omega \mid z = a \wedge a \text{ 为有穷集}\} = \{a\}, \cdots\} \\
M'_\omega = \{M_\omega, \varnothing, \{a\}, a, \cdots (a \text{ 为任意的有穷集})\}
\end{cases}
$$

$\vdots$

从定义易见：M_α，L 均传递，且 $M_\alpha \in L$。

现在，解释一下 L 的直观涵义。

1）L 是从 ϕ，w 出发，经 AB，AS，AP，…等各种运算而得到的所有的结果；且要注意的，这各种运算不是进行任意有穷次，而是进行超穷次运算。

2）L 一方面是具体定义（构造）的 V，L 又依赖于 V，故 L 可写成 L_V；这种依赖性表现在两方面：一是 ϕ 和 ω；二是序数；对不同的 V，构造相应的 L_V 时，要"耗尽"V 中的序数，进行超限次运算（即超穷次运算）。

3）对 ZF 的极小模 M，有相应的 L_M，且 $L_M = M$；在构造 L_M 时只是耗尽 Cantor 意义下的可数序数。

4）在构造 L_V 时，如果令 $M_0 = \omega \cup \{a\}$，其中 a 为极小模 M 中所没有的 ω 的子集合（注意到 ω 有不可数多个子集，而极小模 M 中只有可数个，故这样"设 a 为…"是可以的），而之后与上面同样地进行构造，则所得的相应的 $L_{V(a)}$，也是满足 ZF 公理的。

上述对 L 的解释，可用下面的两个图（图5-2），形象地加以说明。

定理3.2 L 满足 ZF 公理。亦即 L 是一个具体构造（或定义）的 V。

证明 应该说，从上述对 L 的直观解释看，定理显然成立。但

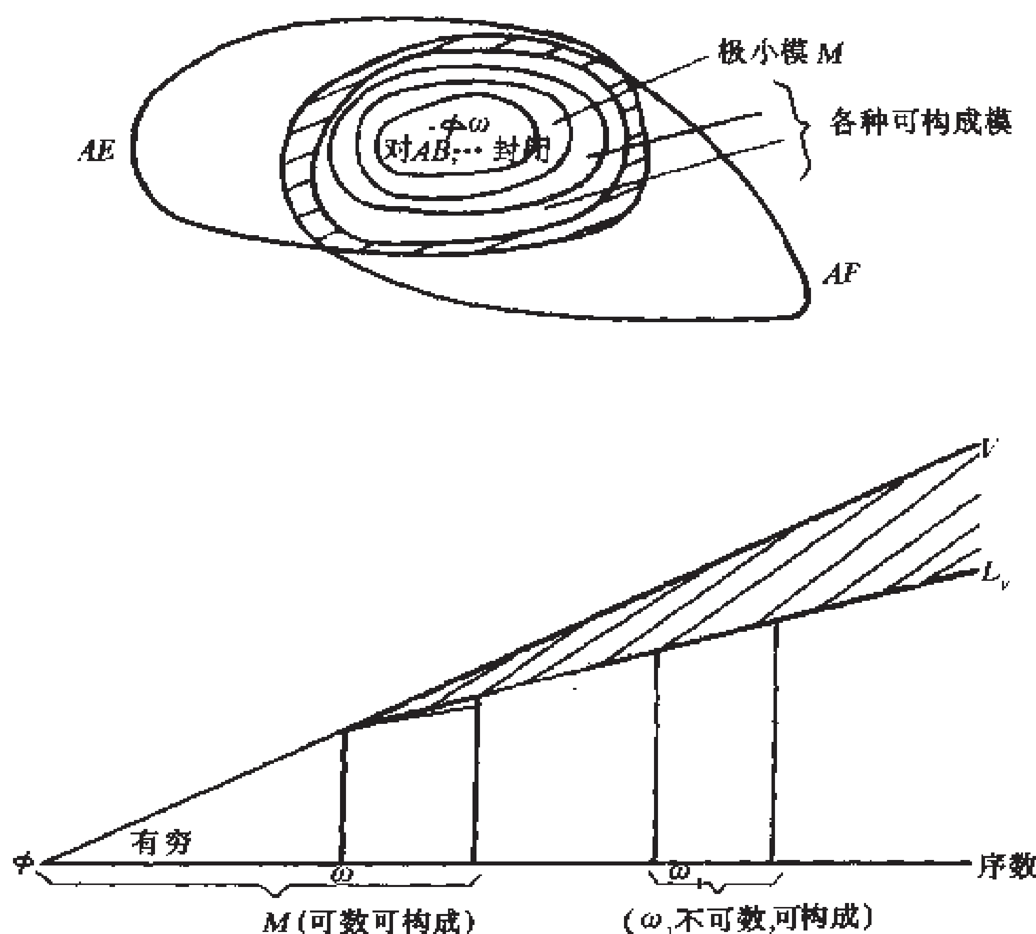


图5-2

为告诉读者如何作类似证明，兹选四条公理严格论证之。这里提醒读者注意，因为是内模型，故对“∈”无需作解释。

1）$A\varnothing$ 在 L 中成立：∵ $\varnothing ⓔ L$，且任 $x ⓔ L$，均有 $\neg x \in \varnothing$（即 $x \notin \varnothing$）；严格写，即

$$\exists y(y ⓔ L \wedge \forall x ⓔ L(\neg x \in y)),$$

故 $A_\varnothing$ 在 L 中为真。

2）Ae 在 L 中成立：即要证

$$\forall x, y ⓔ L(\forall z ⓔ L(z \in x \Leftrightarrow z \in y) \Rightarrow x = y)。$$

∵ L 满足传递性，即

$x ⓔ L \wedge y \in x \Rightarrow y ⓔ L$（由 $x' \subset P(x) \cup x$，且 $M_0 = \varnothing$ 易知）；

∴ $\forall x, y ⓔ L$，在前提 $\forall z ⓔ L(z \in x \Leftrightarrow z \in y)$ 成立时，由 L 的传递

性，对任何 $z\in x, z\in y$，均有 $z\textcircled{\in}L$，故 x, y 的“元素”均在 L 中（这里的“元素”，实际是分别与 x, y 满足关系 $\in$ 的 L 中的个体），因而 x 和 y 的“元素”完全一样，故 $x=y$。（为使读者更容易掌握这里的证明，我们从反面再看看情况会如何。当 L 不满足传递性时，比如说，有 $z\in x$，而 $z \textcircled{\notin} L$，那一当 $\neg z\in y$ 时，就不会有 $x=y$，而前提只是

$$\forall z\textcircled{\in}L(z\in x\Leftrightarrow z\in y)\text{成立},$$

$\therefore Ae$ 就不能保证成立了。）

3）AU 在 L 中成立：

设 $x\textcircled{\in}L$，从 L 定义，就有 α，α 为序数（写作 $O_n(\alpha)$），且 $x\textcircled{\in}M_\alpha$，再由 M_α 传递，任 $y\in x, t\in y$ 均有 $y, t\textcircled{\in}M_\alpha$，那么 $\{t\textcircled{\in}M_\alpha \mid \exists y(y\textcircled{\in}M_\alpha \wedge y\in x \wedge t\in y)\}$ 即为 $\bigcup x$，而其显然也属于 $M_{\alpha+1}$，故 $\bigcup x\textcircled{\in}L$。

4）AB 在 L 中成立：

设 $x, y\textcircled{\in}L$，由 L 定义，就有 $\alpha, \beta, O_n(\alpha), O_n(\beta)$，且 $x\textcircled{\in}M_\alpha, y\textcircled{\in}M_\beta$，再由序数和 M_α 的定义，可设 $\alpha\leqslant\beta$，那么，$x, y\textcircled{\in}M_\beta$，但是

$\{z\textcircled{\in}M_\alpha \mid z=x \vee z=y\}$ 即为 $\{x, y\}$，而其在 M_β^{+} 中，因而，也在 M'_β 中，从而在 $M_{\beta+1}$ 中，故 $\{x, y\}\textcircled{\in}L$。

其他公理，如 $AF, A\omega, AP, AR$ 在 L 中也是成立的，不过证明细节相当繁琐，故这里略去。有兴趣的读者请参阅 Cohen 或 J. L. Krivine 的书。

（3）可构成公理 $V=L$

该公理的涵义是：任何集合均可由构造 L 的过程得到。亦即集合全域 V 等于 L。该公理可由 ZF 的一个公式 A 表示，即 $A\in L_1^S, S=\{\in\}$，而 A 的涵义是 $V=L$。为简单明了起见，常记作 $V=L$。

定理3.3 $V=L$ 在 L 中也成立，即 $(V=L)_L$ 为真。

证明 $V=L$ 可用公式 $\forall x[\exists\alpha(O_n(\alpha)\wedge x\in M_\alpha)]$ 来表示（$\because L$ 是内模型，自然有 $L\subset V$，$\therefore$ 该公式即表示 $V\subset L$）。故 $(V=L)_L$ 即为：

$$\forall x\textcircled{\in}L\exists\alpha\textcircled{\in}L(On(\alpha)\wedge(x\in M_\alpha)_L)$$

亦即 $\forall x \in L \exists \alpha \in L(O_n(\alpha) \wedge \exists y \in L(x \in y \wedge (y = M_\alpha)_L))$,
现在即证明此公式为真。

∵对任 $x \in L$,按 L 的定义,均有 $\alpha, O_n(\alpha)$,且 $\alpha \in M_\alpha$,那么,使用 $\forall x \in L(O_n(x) \leftrightarrow (O_n(x))_L)$(该结果在证 $A\omega$ 在 L 中成立时,要证的。这里,当然均略去),故 $\alpha \in L$;而由 L 的定义,有 $y \in L$,使 $(y = M_\alpha)_L$ 成立。这样,只要证得 $x \in y$ 即全部证毕。而这又要利用"$y = M_\alpha$"的所谓纯粹性(absolute)(所谓"纯粹的"直观涵义是:A 纯粹,则要判定 A 真与否,只要找一个充分大的传递集合,使其足以包含 A 所谈论的对象,那么,在这传递集中判断 A 真与否就足够了)。由此,从 $(y = M_\alpha)_L$ 可得 $y = M_\alpha$(即在 V 中成立),故由 $x \in M_\alpha$,可得 $x \in y$。这样,便全部证毕。

(4)*ZF* 与 *AC*、*GCH* 协调

定理3.4 1) $ZF \vdash (V = L) \rightarrow AC$;

2) $ZF \vdash (V = L) \rightarrow GCH$。

后面再来证**定理 3.4**。现在来看看,如果该定理已证,则得到了重要的结论:ZF 与 AC,与 GCH 均协调。因为假设**定理 3.4** 已证,则对 ZF 的任何模型,$(V = L) \rightarrow AC$,$(V = L) \rightarrow GCH$ 在其中也均成立。再由**定理 3.2**,可构成模 L 是 ZF 的模型,而又由**定理 3.3**,$(V = L)$在 L 中也成立,故 AC,GCH 在 L 中也均必定成立,亦即 $(AC)_L$,$(GCH)_L$ 均为真。这即 ZF 与 AC、GCH 协调。

现在即来证明**定理 3.4 1)**。要证 $(V = L) \rightarrow AC$,即设 $V = L$,来证 AC,而利用 AC 的等价形式 AC_{II}(即良序定理),只需证 L 中的集合均可良序。为此,先证下述**引理**:ZF 中有公式 $A(u, v, x, y)$,使得若 y 是 x 的良序时,则 $u < v$(定义为 $A(u, v, x, y)$)良序 x'。现在来证该**引理**。

定义关系 $C(u, n, t_1, \cdots, t_{k_n}) \equiv u = \{z \in x \mid A^{\,n}_{\,x}(z, t_1, \cdots, t_{k_n})\} \wedge t_i (i = 1, \cdots, k_n) \in x \wedge n \in \omega$;

由 y 是 x 的良序,即下述关系成立:

$\forall t(t \in y \Leftrightarrow t \in x) \wedge \exists R(\forall t \in R \rightarrow t \in x^2) \wedge$($y$ 对 R 是传递的)$\wedge$(y 对 R 满足三歧性);

由此，即可按字典序良序$\underbrace{\langle n,t_1,\cdots,t_{k_n}\rangle}_{k_n+1\text{序组}}$，$n\in\omega,t_i\in x,t_i\in y,i=1,\cdots,k_n$；故对$u\in x^+$，定义

$$\varphi(u)=\mu(k_n+1)\text{序组}[\text{对某 }k_n,C(u,\varphi(u))\text{ 成立}],$$

注意$\varphi(u)$是k_n+1序组，故$C(u,\varphi(n))$为$u=\{z\in x\mid A_x^n(z,t_1,\cdots,t_{k_n})\}$，

那么，这φ即将x^+的元素与最小的构造u的(k_n+1)序组对应起来(按w和y的序最小)。因而，

$$\text{对 }u,v\in x',\text{定义 }u<v=\begin{cases}y\text{ 中序，当 }u,v\in x\text{ 时，}\\ u<v,\text{当 }u\in x,v\notin x\text{ 时，}\\ \varphi(u)<\varphi(v),\quad\text{否则(实即，当 }u,v\in x^+,\\ \qquad\text{而 }u,v\notin x\text{ 时)；}\end{cases}$$

易知，这"$<$"也良序了x'。上述过程是可在ZF中进行的。故**引理**证毕。

现在，利用该**引理**，利用超限归纳法定义M_α的序，从而可完成**定理3.4 1**〉的证明。设对$M_\beta(\beta<\alpha)$的序均已定义好，再按**引理**，定义$(M_\beta)'$的序，而后按自然的次序定义$\bigcup\limits_{\beta<\alpha}(M_\beta)'$的序，即：对$x\in M_\alpha=\bigcup\limits_{\beta<\alpha}(M_\beta)'$，令$f(x)=\mu\gamma[x\in M_\gamma]$，那么，对任$x,y\in M_\alpha$，

$$\text{定义 }x<y=\begin{cases}\text{按 }M_\beta\text{ 中序，}\quad\text{当 }f(x)=f(y)=\beta<\alpha\text{ 时，}\\ x<y,\qquad\text{当 }f(x)<f(y)\text{时。}\end{cases}$$

这样，即可将任何$x\in L$均排好了序。至此，**定理3.4 1**〉证毕。

为证**定理3.4 2**〉，我们不加证明地使用下述二引理：

引理1 $O_n(\alpha)\Rightarrow\alpha\in M_{\alpha+1}$(这在证$A\omega$在$L$中成立时也用，未证明)；

引理2 $x\in M_\alpha,O_n(\alpha),w\leqslant\alpha,y\subseteq x,\Rightarrow\exists\beta,O_n(\beta),\overline{\overline{\beta}}=\overline{\overline{\alpha}},y\in M_\beta$。

现在来证**定理3.4 2**〉。类似证**定理3.4 1**〉，不过这里不直接使用$V=L$，而是使用**1**〉的结果，即使用AC。这是因为已证$V=L\rightarrow AC$，而要证2〉，即假设$V=L$，证GCH；故从$V=L$及1〉，即可使用AC。首先，使用**引理1**，易证$\overline{\overline{\alpha}}=\overline{\overline{M_\alpha}}$($\alpha$无穷时)。因为，当注意到用来定

义 M_α 的元素的公式，故 $\overline{\overline{M_\alpha}} \leqslant \overline{\overline{\alpha}}$；而由**引理1**，$\alpha \in M_{\alpha+1}$ 且 $M_{\alpha+1}$ 又传递，故 $\overline{\overline{\alpha}} \leqslant \overline{\overline{M_\alpha}}$。因而，得 $\overline{\overline{\alpha}} = \overline{\overline{M_\alpha}}$。其次，设 α 为基数，且 β 为 α 的下一个基数，由**引理1**，$\alpha \in M_{\alpha+1}$；由**引理2**，对任何 $y \subseteq \alpha$，均 $\exists \gamma, O_n(\gamma)$，$\overline{\overline{\gamma}} = \overline{\overline{\alpha + 7/8}}$，$y \in M_\gamma$；再由 M_β 传递，可得 $P(\alpha) \subset M_\beta$，$\therefore \overline{\overline{P(\alpha)}} \leqslant \overline{\overline{M_\beta}} = \overline{\overline{\beta}}$；但从 Cantor 定理，$\overline{\overline{P(\alpha)}} > \overline{\overline{\alpha}}$，$\therefore \overline{\overline{P(\alpha)}} \geqslant \overline{\overline{\beta}}$；结合上述，得到

$\overline{\overline{P(\alpha)}} = \overline{\overline{\beta}}$ 而 β，即 α 的下一基数，这即 GCH。

因而，**定理3.4 2〉**也证毕。

3.5 *ZF* 的其他模型，*ZF* 与 $V \neq L$，$\neg AC$，$\neg CH$ 的协调性

(1)三条更高的公理：即模型存在公理 AM，标准模型公理 ASM 和大基数(不可达基数)公理 AI。

AM：有一集合 $\mathbb{M}$ 和一个二元关系 $\in \subseteq \mathbb{M}^2$，其是 ZF 的模型。可用 ZF 句子

$\exists x(A\phi_x \wedge Ae_x \wedge AU_x \wedge AB_x \wedge AP_x \wedge AS_x \wedge AR_x \wedge AF_x \wedge A_{\omega_x})$

来表示它。当然，对公理模式 AS 和 AR，则要将公式的形成规则也用 ZF 的语言表述，方能作到。从第三章中，在 Peano 算术中描述公式的规则可知，这是不难作到的。

ASM：有一集合 $\mathbb{M}$，使如果 $R = \{\langle x, y\rangle \mid x \in y \wedge x \in \mathbb{M} \wedge y \in \mathbb{M}\}$，则 $\mathbb{M}$ 是 ZF 的模型。该公理使“$\in$”和“属于”建立关系，这是不同于 AM 的。

AI：有一不可数基数 A，使得1〉如果 $B < A$，则 A 不是小于 A 的这些 B 之和，2〉对任何 $B < A$，$\overline{\overline{P(B)}} < A$。由于所有的秩小于 A 的集合是 ZF 的模型，故 AI 也蕴涵着 ZF 有模型。

鉴于这三条公理均蕴涵着 ZF 有模型，从而也蕴涵着 ZF 协调，故由 Gödel 不完全性定理(即 Gödel 第二不完全定理：Peano 算术协调是独立于 Peano 算术的)，$ZF \nvdash AM, \neg AM, ASM, \neg ASM, AI, \neg AI$。这样，$ZF$ 加上 AM 或 ASM 或 AI，才是真正加入了新的公理。

(2)极小模 M(见图5-3)：

在介绍可构成模 L 时，就讲到 ZF 的极小模 M。任何 ZF 的模

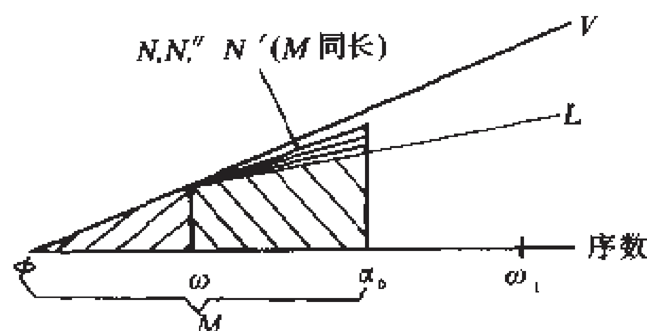


图5-3

型都包含 M。那么有：

定理 3.5 $ZF+ASM$ 蕴涵着：有一个唯一的可数传递模 M，使如果 N 是一标准模，则有 M 到 N 的 $\in$-同构。

(3)Cohen 构造模型的基本思想：先证一定理，用内模型法构造不出 $ZF+V\neq L$，$ZF+\neg AC$，$ZF+\neg CH$ 的模型。由此，要获得上述结果，只能另谋他途，比如“外模型”法。但他仍采用 Gödel 构成 L 时的直谓的(即分成层次的)方法，通过加宽极小模 M，得到三个相应的模型 N，N''和 N'。如图5-3所示。

因为是加宽 M，将一些新对象加入到 M 中，故必须对这些新对象间，及对新、老对象间解释“$\in$”和“$=$”。这不像内模型法，都是原来 ZF 中的，因而也无需作解释。

比如 Cohen 构造模型 N 时，是令 $M_0=\omega\cup\{\bar{a}\}$，之后构造办法同极小模 M。但要将 $\bar{a}$ 解释为不可构成的集合，即得到 $V\neq L$ 在其中成立。又如他构造的模型 N'，则是往 M_0 中加入函数，使达到 $\neg CH$ 在模型中成立。而当他构造 N''时，则是往 M_0 中加入不能良序的对象，使达到 $\neg AC$ 在其中成立。

当往 M_0 中加入新的个体(比个 $\bar{a}$ 等)时，可用公理刻划它们，也可不加公理，采取另外的方法。Cohen 则是通过他所创造的力迫条件来刻划这些新对象，从而使满足所需性质的新对象，被强行加入论域之中，然后再解释“$\in$”、“$=$”等关系，从而证明公理在新模

型中成立。

这里解释一下所谓力迫条件。力迫条件就是形如 $\overline{n}\in\overline{a}$，$\overline{m}\notin\overline{a}$（$\overline{m},\overline{n}\in\overline{\omega}$）的公式组成的协调的有限集，Cohen 使用无限个力迫条件来刻划新加入的对象。比如，对 $\overline{a}$ 解释为自然数的子集合 a，亦即使得

$$a=\{k\mid(\exists n)(\overline{k}\in\overline{a}\text{ 在 }P_n\text{ 中}\wedge Pn\text{ 为力迫条件})\};$$

而解释"∈"、"="则需定义力迫关系，即

$$c\in c'\ \overline{\overline{df}}(\exists n)P_n\Vdash\overline{c}\in\overline{c'}$$

$$c=c'\ \overline{\overline{df}}(\exists n)P_n\Vdash\overline{c}=\overline{c'},$$

其中，"⊩"为力迫关系符号。Cohen 书中用19条规则来定义力迫关系。其基本思想有二：1）凡极小模 M 中的个体应同原来的；2）凡涉及新加入的对象，则应符合力迫条件的信息中所含之内容。

最后，在构造模型中特别要注意有无带入引起麻烦的东西。比如有无新的序数，有无引入和所需性质相矛盾的东西。王浩在他的书中评论 Cohen 的模型时，说："Cohen 的方法给出了一种以很经济的方式增加新的集合来扩充模型的方法。它使得新的模型仅有为增加新集合所力迫的性质，而使新集合与模型的相互影响保持到必要的最小限度。"

练 习 题

第二章 理想计算机与有穷性原则

§1. 递归函数

(1)试证明 $x^y, x \dot{-} y, |x-y|, Sg(x), \overline{Sg}(x), rem(x,y), [x/y], div(x,y)$

等函数均是原始递归函数。

(2)试证明连加、连乘是原始递归算子，即，如果$f(x_1,\cdots,x_n,z)$是原始递归函数，则

$$g(x_1,\cdots,x_n,y)=\sum_{z=0}^{y} f(x_1,\cdots,x_n,z)$$

和

$$h(x_1,\cdots,x_n,y)=\prod_{z=0}^{y} f(x_1,\cdots,x_n,z)$$

也是原始递归函数。

(3)试证明分情形定义算子是原始递归算子

*(4)若$f(n)=u_n$是斐波那契数列，即

$$u_0=u_1=1, \text{对} n\geq 2, un=un=u_{n-1}+u_{n-2},$$

试证明 f(n)是原始递归函数。

$$\left(\text{提示}: u_n=\frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n+1}-\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}}{\sqrt{5}}\right)$$

(5)试证明$g(x)=\begin{cases}0, \text{如果 } x \text{ 为偶数}; \\ \text{无定义}, \text{如果 } x \text{ 为奇数};\end{cases}$

是部分递归函数。

§3. Turing 机器

(1)试构造 Turing 机计算以下函数：

(a)加法函数：$Add(x,y)=x+y$

(b)恒等函数:$E(m,n)=\begin{cases}1,如果\ m=n\\0,如果\ m\neq n;\end{cases}$

(2)试构造 Turing 机计算乘法函数 $x\cdot y$。

(3)按照3.3中所提供的方法,具体构造一个符号集为{0,1,}的通用 Turing 机。若不假设 T' 的符号集为{0,1},会有什么不同?

(4)试构造二状态,三状态忙海狸计算 $\Sigma(2)$,$\Sigma(3)$。

(5)思考:为什么计算机是有穷自动机?有穷自动机不能计算乘法,那计算机的"乘法"是怎样实现的?

(6)如果 $f(x_1,\cdots,x_n)$是 Turing 机可计算函数,试证明有无穷多个不同的 Turing 机计算它。

(7)给出一个能行的编码函数 $\ulcorner\ \urcorner$,其满足:

1)若 M_1,M_2是两个不同的 Turing 机,则 $\ulcorner M_1\urcorner \neq \ulcorner M_2\urcorner$;

2)对任一 Turing 机 M,可以能行地找到自然数 m,使 $m=\ulcorner M\urcorner$;

3)对任一自然数 m,可以能行地判定是否有 Turing 机 M 使 $m=\ulcorner M\urcorner$,并能行地找到相应的 M。

*(8)Turing 机停机问题:

假设 $\ulcorner\ \urcorner$ 是一个满足上题1),2)和3)的编码函数,那么,令

$$h(m)=\begin{cases}0,如果有\ \ulcorner M\urcorner=m,且\ M\ 对输入空白带停机;\\1,否则。\end{cases}$$

试证明 h 不是 Turing 机可计算函数。

(提示:利用 $\Sigma(n)$的不可计算性,并利用递归函数与 Turing 机的等价性。)

(9)证明 Ackermann 函数是 Turing 机可计算函数。

(10)利用递归函数等价于 Turing 机可计算函数,并设 g(n)=e 的小数表示的第 n 位数字,即

$$g(0)=2,g(1)=7,g(2)=1,g(3)=8,\cdots$$

证明 g(n)是 Turing 机可计算函数。

*(11)设 $G(f)$是函数的图象,即

$G(f)=\{\underline{x_1}0\underline{x_2}0\cdots0\underline{x_n}00\underline{f(x_1,\cdots,x_n)}\mid f(x_1,\cdots,x_n)$有定义},其中,$\underline{n}$表示 $n+1$个"1"的序列。若有 Post 系统$(A,\{0\},P)$,$0,1\in A$,使 $G(f)$是它的导出集,则称 f 是 Post 可计算函数。试证明:Post 可计算函数与 Turing 机可计算函数等价。

§ 4. 计算机的数学模型(MMCM)

(1)试编制计算十进制乘法的 MMCM。

(2)模仿上节 Turing 机可计算函数与递归函数的等价性的证明，证明：URIM 可计算函数与递归函数等价。

(3)设 A,B 是自然数集 N 的子集，定义

$$A\oplus B=\{2x\mid x\in A\}\cup\{2x+1\mid x\in B\},$$

试证明 $A\oplus B$ 是递归集当且仅当 A,B 均递归。

(4)设 f 是一元 Turing 机可计算函数，$A\subseteq \mathrm{Dom}(f)$，令 $g=f\upharpoonright A$，证明 g 是 Turing 机可计算当且仅当 A 递归可枚举。

(5)设 f 是一元函数。试证明：f 是 Turing 机可计算的当且仅当 $\{2^x3^{f(x)}\mid x\in \mathrm{Dom}(f)\}$ 是递归可枚举集。

(6)设 A 是无穷递归可枚举集，证明：A 可被一个全定义递归函数不重复地枚举。

(7)设 f 是全定义递归函数，而 A 是递归集，B 是递归可枚举集。证明 $f^{-1}(A)$ 是递归集，而 $f(A)$，$f(B)$ 和 $f^{-1}(B)$ 是递归可枚举集，但不必是递归集。如果 f 是一个1－1对应的双射，有什么不同？

(8)设 A,B 是丢番图集，试证明：$A\cup B$，$A\cap B$，$A\oplus B$(见题(3)定义)均是丢番图集。

第三章　有穷性逻辑和有穷性数学

§ 1. 数理逻辑和数学

(1)试构造两个 Post 系统分别生成本节中群语言的项及合式公式之集。并证明这二个集合是递归集。

(2)试从直观上验证本节中各条推理规则的可靠性。

§ 2. 一阶逻辑

(1)若 A 至多可数，证明 A^* 可数。

(2)论述并证明公式的唯一分解定理。

(3)令 $S=\{+,\cdot,0,1\}$，试给出一组公式刻划 Peano 算术。

(4)试证明以下各推理关系：

1) $A\to B\vdash\dashv\neg A\vee B$;

2) $A\leftrightarrow B \vdash\dashv (A\rightarrow B)\wedge(B\rightarrow A)$；

3) $\neg(A\wedge B)\vdash\dashv\neg A\vee\neg B$，$\neg(A\vee B)\vdash\dashv\neg A\wedge\neg B$（德·摩根律）；

4) $A\wedge(A\vee B)\vdash\dashv A$，　　$A\vee(A\wedge B)\vdash\dashv A$（吸收律）；

5) $A\wedge(B\vee\neg B)\vdash\dashv A$，　　$A\vee(B\wedge\neg B)\vdash\dashv A$；

6) $\left.\begin{array}{l}A\wedge(B\vee C)\vdash\dashv(A\wedge B)\vee(A\wedge C)\\ A\vee(B\wedge C)\vdash\dashv(A\vee B)\wedge(A\vee C)\end{array}\right\}$　（分配律）。

(5)利用上题证明合取(析取)范式定理：任一公式 A 均有与其等价的合取(析取)范式。

(6)写出以下公式的合取范式和析取范式：

1) $((A\rightarrow A\vee B)\rightarrow B\wedge C)\leftrightarrow\neg A\vee C$；

2) $(A\leftrightarrow B)\leftrightarrow((\neg A\leftrightarrow C)\rightarrow(B\leftrightarrow\neg C))$

(7)试证明以下推理关系：

1) $\neg\forall xA(x)\vdash\dashv\exists x\neg A(x)$，$\neg\exists xA(x)\vdash\dashv\forall x\neg A(x)$；

2) $\forall xA(x)\wedge\forall xB(x)\vdash\dashv\forall x(A(x)\wedge B(x))$，

$\exists xA(x)\vee\exists xB(x)\vdash\dashv\exists x(A(x)\vee B(x))$；

(8)设 Qx，Q_1x，Q_2x，是 $\forall x$ 或 $\exists x$，试证明：

1) $A\wedge QxB(x)\vdash\dashv Qx(A\wedge B(x))$，

$A\vee QxB(x)\vdash\dashv Qx(A\vee B(x))$；

2) $Q_1xA(x)\wedge Q_2yB(y)\vdash\dashv Q_1xQ_2y(A(x)\wedge B(y))$，

$Q_1xA(x)\vee Q_2yB(y)\vdash\dashv Q_1xQ_2y(A(x)\vee B(y))$；

3) $QxA(x)\vdash\dashv QyA(y)$。

(9)利用上述(4)，(7)，(8)，证明前束范式定理：任一公式 A，均有与其等值的前束范式。

(10)由定理5，证明定理4a：若对任意 Φ，当 Φ 可满足时，都有 Φ 协调，则对任意 Φ 和 φ，$\Phi\vdash\varphi$ 都是正确的。

*(11)令 $\Sigma=\{\neg,\wedge,\vee,\rightarrow,\leftrightarrow,\exists,\forall,\equiv,(,>,v,c,1,0,f\}$，并以 v，$v1$，$v11$，…表示变元 v_0，v_1，v_2，…

c，$c1$，$c11$，…表示个体词 C_0，C_1，C_2，…

则项，公式、证明等都可以看成是 Σ^* 中的元素。

试证明关系：$\mathscr{A}(x,y)$当且仅当 x 是公式，y 是 x 的形式证明是 $\Sigma^*\times Z^*$ 上的递归关系。(提示：先证明项、公式、推理规则之集皆是递归集。)

(12)令 $S=\{+,0,1\}$，而 Φ 为公式集：

$\forall x\neg x+1=0$，

$\forall x \forall y(x+1=y+1\to x=y)$,

$\forall X((X0 \wedge \forall x(Xx\to X\, x+1))\to \forall x Xx)$,

(注意：第三条是二阶公式)且设$N\models\Phi$；试证明：Φ 的任一模型都与N 同构。

§4. 有穷和无穷命题演算

(1)试给出P_w中关于$\to$和$\leftrightarrow$的推理规则($l4a$)~($l5b$)

(提示：$\Gamma_1 \vdash_w \Gamma_2$等价于$\Gamma_1 \vdash \vee \Gamma_2$等价于

$$\vdash \neg\, p_1 \vee \cdots \vee \neg\, p_m \vee q_1 \vee \cdots \vee q_m$$

而 $\Gamma_1=\{p_1,\cdots,p_m\}$，$\Gamma_2=\{q_1,\cdots,q_m\}$)

(2)试按照P_w的推理规则，证明：

$$\vdash_w((p \vee \neg\, q) \wedge q)\leftrightarrow(p \wedge q)。$$

(3)就有穷命题演算的"形式证明"、"语义"、"正确性"、"完全性"，叙述有关的定义，陈述有关的定理，并证明之。

(4)陈述有穷命题演算的标准合取范式定理，并给出证明。

(5)陈述无穷命题演算的标准合取范式定理，并给出证明。

第四章　一般逻辑和一般数学

§1. 一般逻辑和一般数学的定义

(1)证明：1) $\bigwedge_{t\in T} p_t \Vdash p_{t_0}$，$t_0\in T$；

2) $\neg(p_0\wedge\neg\, p_1)\wedge\neg(p_1\wedge p_2)\wedge\neg(p_0\wedge\neg\, p_2)\Vdash\neg\, p_0$。

(2)证明：$p_0\to p_{t_0}\Vdash p_0\to\bigwedge_{t\in T} p_t$，其中$0\notin T$，而 $t_0\in T$。

第五章　集　合　论

§1. 朴素集合论

(1)试证明任何可数的全序集 A，可在有理数集 Q 中找到全序子集 Q_0，使 $A\simeq Q_0$。

(2)证明全序和的结合律：$(\mu+\gamma)+\xi=\mu+(\gamma+\xi)$。

(3)设 D 的序型是 γ，定义全序积：

$$\mu\cdot\gamma=\sum_{\lambda\in D}\mu;$$

证明全序积的结合律及对全序和的右分配律：

$$(\mu\cdot\gamma)\cdot\xi=\mu\cdot(\gamma\cdot\xi);$$

$$\mu\cdot(\gamma+\xi)=\mu\cdot\gamma+\mu\cdot\xi。$$

(4)证明序数 $\mu+1$是 μ 的直后序数。

*(5)令 W_1表示所有的可数全序序型，试证明$\overline{\overline{W_1}}=C$。由此，连续统猜想等价于：可数良序序型与可数全序序型的势相等。(提示：对 $A\subseteq N$，若 $n\in A$，则 $A_n=N$；反之，$A_n=N^*$，令 $\xi_A=\sum_{n\in N}A_n$，证明若 $A\neq B$，则 $\xi_A\neq\xi_B$。)

§2. 公理集合论

(1)设 S_1,S_2是集合，证明 $S_1\times S_2=\{\langle x_1,x_2\rangle\mid x_1\in S_1,x_2\in S_2\}$也是集合。

(2)试以一阶语言描述：$y\cup\{y\}\in x$。

(3)试证明：任一序数均是传递的。

*(4)若 α,β 是二序数，试证明：

$$\alpha\in\beta \text{ 或 } \alpha=\beta \text{ 或 } \beta\in\alpha \text{ 必有一成立。}$$

(提示：$\alpha\cup\beta$ 是序数，满足三歧性。)

(5)由正则公理 AF 知：每一序数 α 都是$\in$-良序的，即关系$\in$是 α 上的一个良序。若不用 AF，试证明：

1)0是$\in$-良序的；

2)若 α 是$\in$-良序的，则 α^+是$\in$-良序的；

*3)若 S 是序数的集合，则$\cup S$ 是$\in$-良序的。

(提示：利用(4))

*(6)设 A 是一集合，利用 AR 证明：

$$W_A=\{\beta\mid \text{存在 1—1 函数 } f:\beta\to A\} \text{ 是一个序数。}$$

§3. ZF 系统

(1)试证明 $AC_{\text{V}}\Rightarrow AC_{\text{VI}}$，即势的三歧性蕴含良序定理。(提示：利用上节的练习(6)，即$\overline{\overline{A}}\leqslant\overline{\overline{W_A}}$。)

(2)试证明 $AC\,\text{VI}\Rightarrow AC\,\text{VII}$，即良序定理蕴含选择公理。(提示：利用$\cup S$ 上的良序。)

(3)设$(V,Ⓢ)$满足 ZF 公理，f 是 V 到 V 上的1-1对应，且有一个一阶公式 $A(x,y)$，使 $f(x)=y$ 当且仅当 $A(x,y)$；令 $x\epsilon y$ 当且仅当 $x\in f(y)$；则有：(V,ϵ)，满足 ZF 中除 AF 外的每条公理。

*(4)利用上题，构造一个(V,ϵ)，满足 ZF 中除 AF 外的每条公理，并且有 $a\in V$，使得 $a\epsilon a$。

参考文献

[1]张鸣华,可计算性理论,清华大学出版社,1984

[2]胡世华、陆钟万,数理逻辑基础(上、下册),科学出版社,1981,1982

[3]王世强,模型论基础,科学出版社,1987

[4]唐稚松,逻辑网络(讲义),1960

[5]张锦文,集合论浅说,科学出版社,1984

[6]张锦文,公理集合论导引,科学出版社,1991

[7]涂克仁,集合论和近世代数(讲义),1960

[8]方嘉琳,集合论,吉林人民出版社,1982

[9]谢邦杰,超穷数与超穷论法,吉林人民出版社,1979

[10]肖文灿,集合论初步,商务印书馆,1939,1950(再版)

[11]王浩,数理逻辑通俗讲话,科学出版社,1981

[12]王宪钧,数理逻辑引论,北京大学出版社,1982

[13]华罗庚,数论导引,科学出版社,1979

[14]关于忙海狸函数,科学,1984. No. 7(英文原文刊于 Scientific American, Feb. 1984)

[15]徐书润　孙生录,有穷自动机所计算的函数,数学进展,1966. 9卷9期

[16]徐书润　王永革,加速定理与函数分层,软件学报,1993. 4 No. 4.

[17]徐书润,G 函数分层(手稿),1965

[18]乔海燕　徐书润,一类完全递归函数的分层,数学季刊,1987

[19]徐书润,计算与机器计算,大自然探索,1987

[20]徐书润,有穷逻辑的数学理论的 Gödel 不完全性(手稿),1989

[21]陈奇,一致可分 MMCM 的能力的分析(硕士论文),1988

[22]周志东,从串行一致可分 MMCM 程序到并行 PL I 程序的自动转换系统(硕士论文),1990

[23]宋　曦,一致可分 MMCM 的功能的进一步研究(硕士论文),1995

[24]Hong Jiawei, Computation: Computability, Similarity and Duality, Pitman, London, 1986

[25]Hu Guoding, Ordinary Mathematics and Formal Mathematics,南开数学所研究报告,No. 1997-M-005

[26]Hu Guoding, On Mathematical Model of Computing Machine, J. of Computer Sci. and Tech., 3: 4, 1988

[27]Hu Guoding, Parallel Computation Simulating Sequential Computation

[28]Xu Shuruen, On the Equivalence of Some Models of Computation, J. of Computer Sci. and Tech. ,3:4,1988

[29]Xu Shurun, Petri Net-like Languages, Fourth Asia Logic Conference (Tokyo), 1991

[30]Wang Kewen & Xu Shurun, New Relation Between The Shift Function and The Busy Beaver Function, Chinese J. of Advanced software Research, 2:2, 1995

[31]Yang Ruiguang, Ding Longyun & Xu Shurun, Some Better Results Estimating the Shift Function in Terms of Busy Beaver Function, ACM SIGACT News, March, 1997

[32]Zhao Jie, Simulating Uniform Separable Computation with Petri Net-like Languages, Fourth Asia Logic Conferenc, (Tokyo), 1991

[33]S. C. Kleene, Introduction to Metamathematics, Van Notrana, Amasterdam, 1952

[34]H. D. Ebbinghaus, Mathematical logic, Springer, 1984

[35]P. J. Cohen, Set Theory and the Continuum Hypothesis, Benjiamin, New York, 1966

[36]N. Cutland, computability, Cambridge Univ. Press, 1980

[37]J. Barwise etc. (ed.), Handbook of Mathematial Logic, North-Halland publishing Company, Amastedam, 1977

[38]J. Barwise etc. (ed.)Model Theoretic Logics, Springer, 1985

[39]R. L. Epstein & W. A. Carnielli, Computability; Conputable Functions, Logic and the Foundations of Mathematics, Wadsworth & Brooks…, American, 1989

[40]A. Grzegorczyk; Some Classes of Recursive Functions, 数理逻辑论文选, 1953

[41]C. B. Dunham, A Sequence of Uncomputable Functions, ACM SIGAT News, 16: 3, 1984

[42]D. S. Johnson, The NP-Completeness Column: An Ongoing Guide, J. of Algorithms, many periodical.

[43]J. Hartmanis, The Structural Complexity Column, Bull. European Asso. Theor. Compt. Sci. every periodical

[44]M. Davis, Y. Matijasevic & J. Robinson, Hilbert's Tenthe Prob. , Diophantine Equations; Positive Aspects of A Negative Solution, Math, Devel. Arising From Hilbert Problems, 1976

[45]S. Shapiro, Foundations Without Foundationalism, Oxford Univ. Press, 1991

[46]H. J. Keisler, Model Theory for Infinitary Logic, North-Holland P. C. , Amasterdam, 1971

[47]H. J. Keisler, Logic with the Quantifier " there exist uncountable many", Annals of

Math, logic vol. 1(pp. 1—93), 1970

[48]P. R. Halmos, Naive Set Theory, 1960

[49]J. L. Krivine, Introduction to Axiomatic Set Theory, Reidel, Dordrecht, 1971

[50]T. T. Jech, Set Theory, Academic Press, New York, 1978

[51]R. M. Smullyan, Recursion Theory for Metamathematics, Oxford Univ, Press, 1993

[52]R. M. Smullyan, Gödel Incompleteness Theorem, Oxford Univ, Press, 1992

[53]R. Kaye, Models of Peano Arithmetic, Clarendon Press, Oxford, 1991

[54]H. Kotlarski, On the Incompleteness Theorem, J. of Symbolic Logic, 59:4, 1994

[55]H. Kotlarski, An Addition to Rosser's Theorem, J. of Symbolic Logic, 61:1, 1996

[56]H, Putnam, Nonstandard Models and Kripkes Proof of the Gödel Theorem, manuscript, 1984

[57]J. Paris & L. Harrington, A Mathematical Incompleteness in Peano Arithmetic, Handbook of Math. Logic pp. 1133—1142, 1977

[58]R. W. Ritchie, Classes of Predictably Computable Functions, Trans. Amer. Math. Soc. Vol. 106(pp. 139—173), 1963

[59]M. L. Minsky, Computation: Finite and Infinite Machines, Prentice-Hall, Englewood Cliffs, N. J., 1967

[60]A. R. Meyer & K. Winklmann, The Fundamental Theorem of Complexity Theory (Preliminary Kersion), MC Tract 108, Foundations of Computer Sci. Ⅲ, J. W. de Bakker (ed.), MIT, USA, 1979

[61]B. A. Julstrom, A Bound on the Shift Function in Terms of the Busy Beaver Function, ACM SIGACT News, pp. 100—106, 1992

[62]R. Machlin, The Complex Behaviour of Simple Machines, Physica D. Nonlinear Phenomena 42:1-3(pp. 85—98), 1990

[63]A. M. Ben-Amram, B. A. Julstrom & U. Zwick, A Note on Busy Beavers and Other Creatures, Math. Systems Theory, Vol. 29(pp. 375—386), 1996